AD-A047 999 AIR FORCE AVIONICS LAB WRIGHT-PATTERSON AFB OHIO F/G 17/2
ROOFTOP ANTENNA POINTING PROGRAM. (U)
NOV 77 P F HUMEL
UNCLASSIFIED AFAL-TR-77-228 NL

1 of 2
ADA047 999

AFAL-TR-77-228

ADA047999

ROOFTOP ANTENNA POINTING PROGRAM

System Development Branch
System Avionics Division

1

November 1977

TECHNICAL REPORT AFAL-TR-77-228

Final Report for Period March 1974 - October 1977

Approved for public release; distribution unlimited

AIR FORCE AVIONICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
Air Force Systems Command
Wright-Patterson Air Force Base, Ohio 45433

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

ALLEN JOHNSON, Project Monitor

CHARLES C. GAUDER
Chief, System Development Branch
System Avionics Division

FOR THE COMMANDER

RICHARD W. SMITH, Lt Col, USAF
Acting Chief
System Avionics Division

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify AFAL/AAD-2 *,W-PAFB, OH 45433 to help us maintain a current mailing list".*

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFAL-TR-77-228	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Rooftop Antenna Pointing Program		5. TYPE OF REPORT & PERIOD COVERED Final March 1974 - October 1977
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Capt Paul F. Humel		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS System Development Branch (AAD) Air Force Avionics Laboratory (AFSC) Wright-Patterson AFB, Ohio		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Project No. 1227 Task No. 122712 Work Unit No. 12271217
11. CONTROLLING OFFICE NAME AND ADDRESS System Avionics Division (AA) Air Force Avionics Laboratory (AFSC) Wright-Patterson AFB, Ohio		12. REPORT DATE November 1977
		13. NUMBER OF PAGES 134
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

LES 8/9 Satellite
Communications
Digital Processing

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This document provides a description of the in-house work performed in the Air Force Avionics Laboratory's Communication Systems Evaluation Laboratory (CSEL) to successfully point a ten-foot dish antenna at any earth orbiting satellite for which ephemeris data is available. Both the hardware and software aspects of the program are covered.

FOREWORD

This technical report was prepared in the System Development Branch (AAD), System Avionics Division (AA), of the Air Force Avionics Laboratory, Wright-Patterson Air Force Base, Ohio. The work was accomplished under Project No. 1227, "Advanced Microwave Communications", Task No. 12, "Communication Test and Evaluation", Work Unit 17, "Avionics Laboratory Rooftop Antenna Pointing Program". The Project Engineers were Lt Paul F. Humel (AFAL/AAD) and Mr. David E. Muench (formerly of AFAL/AAF).

Acknowledgment is given to Mr. Boyd Holsapple (formerly of AFAL/AAF), Ms. Susan Pfaadt and Ms. Audrey Goldsmith of Systems Research Laboratory, whose programming assistance was appreciated, and Mr. Glenn Sellars of Lincoln Laboratory, who designed and built the Crosslink Ranging Receiver.

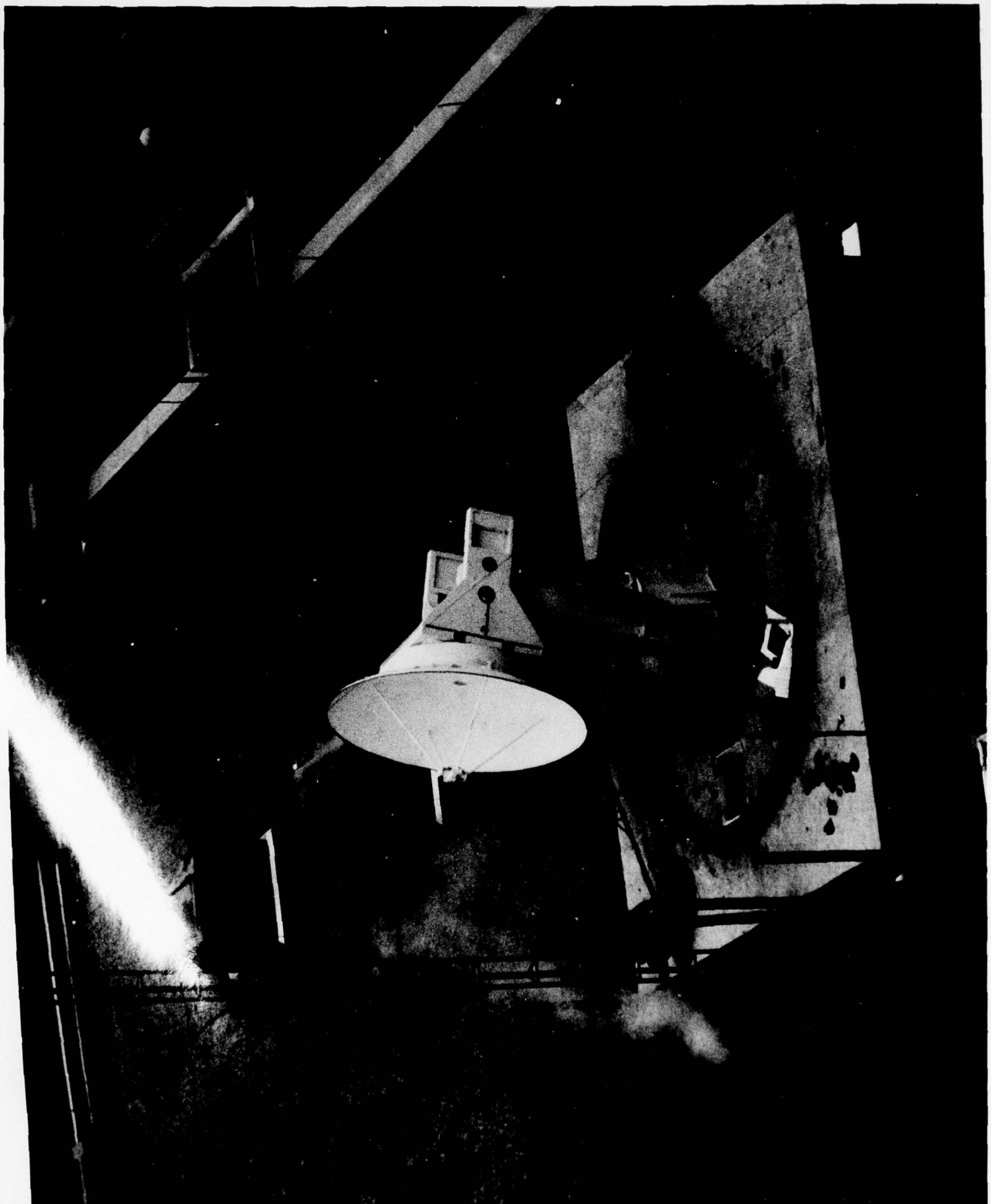


FIGURE 1

TABLE OF CONTENTS

SECTION I

INTRODUCTION

This document consists of the final report prepared in-house by the Air Force Avionics Laboratory, AFAL/AAD, Wright-Patterson AFB, Ohio, under Work Unit 12271217, entitled Avionics Lab Rooftop Antenna Pointing Program.

1.1 Objective of the Program. In March 1976 the Lincoln Experimental Satellites, LES 8 and LES 9, were launched into earth orbit. Operating in the UHF and Ka bands, these communications satellites are the forebearers of the next generation of the AFSAT communications satellites being developed for the Air Force's Advanced Airborne Command Post. The Satellite Communications Group of the Air Force Avionics Laboratory has been tasked with testing of the various satellite configurations. To perform the testing, a complete communication system was installed in the Rooftop Facility atop the Avionics Laboratory, Building 620, Wright-Patterson AFB. A complete system was also installed in a C135 aircraft, so that airborne testing could be performed. The aircraft has a three-foot antenna for K-Band communications, while the Rooftop utilizes a ten-foot parabolic dish antenna, with a 3 db beam width of only two-tenths of a degree. Location and passive tracking of the satellites from the Rooftop, without some form of computer control, would be virtually impossible. It is to meet this need for exact antenna pointing that the Rooftop Antenna Pointing Program was developed.

1.2 Implementation. To point the ten-foot dish, located atop the Rooftop Facility (see Figure 1), a Digital Equipment Corporation PDP 11/45 minicomputer was purchased and interfaced with the antenna's servo control system. The Rooftop Antenna Pointing Program (RAPP) and associated programs were written, and occupy approximately 48,000 words of core memory. RAPP

operates under the RSX 11-D Realtime Operating System, which occupies another 24,000 words of memory, for a total of approximately 72,000 words. Normal CPU loading is only five percent of available CPU time. However, peak loading can run up to 100% for periods of two to three minutes at a time. RAPP is general enough for use with any satellite for which proper ephemeris data is available.

1.3 Contents of This Report. This report consists of four sections. Section II consists of a description of the various programs associated with RAPP. Section III contains listings of the RAPP programs. Section IV consists of an operator's guide and fault finding chart for the RAPP user. Sections II and IV assume that the reader is familiar with FORTRAN, in addition to the instruction set of the Digital Equipment Corp. PDP 11/45. A knowledge of RSX 11-D and DOS is also assumed.

SECTION II

ROOFTOP ANTENNA POINTING PROGRAM DESCRIPTION

2.1 Hardware Introduction. The hardware consists of a PDP 11/45 minicomputer with the following peripherals: 40,000 words of core memory, 32,000 words of MOS memory, 2.4 million words of disk memory, 20 million words of magnetic tape memory, a card reader, a line printer, and a DECWRITER. In addition, a VTØ5 video display terminal is located in the Rooftop Facility. The antenna servos, a Crosslink Ranging Receiver and a WWVB Receiver, are interfaced with the 11/45 via DR11-C general device interfaces. Figure 2 illustrates the hardware configuration.

2.2 Detailed Description of the Hardware:

2.2.1 DEC Hardware. Most of the hardware is standard Digital Equipment Corporation (DEC) equipment, and need not be discussed in detail. The following list provides a breakdown of the DEC equipment:

KB11-A	11/45 CPU
FP11-B	Floating Point Unit
RKØ5	Disk Drives (2 ea)
MS11	32K of MOS Memory
MM11	40K of Core Memory
KW11-P	Real Time Clock
LA36	Data Terminal
VTØ5	CRT Display
CR11	Card Reader
LP11	Line Printer
TV1Ø	Magnetic Tape Unit

In addition to the DEC equipment, three additional pieces of hardware are interfaced with the 11/45 via DR11-C general device interfaces. They are the Antenna Control Interface Unit, the Crosslink Ranging Receiver, and the WWVB Receiver.

2.2.2 Antenna Control Interface Unit. The Antenna Control Interface Unit (ACIU) was built by the Raytheon Company as part of the

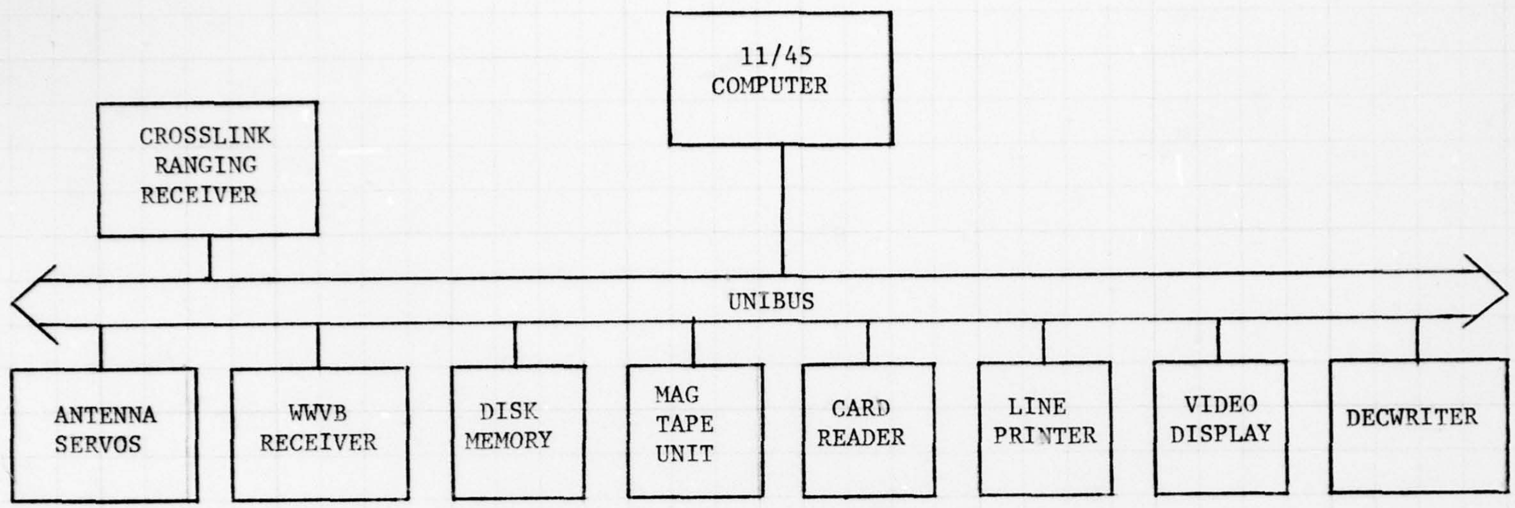


FIGURE 2

RAPP COMPUTER CONFIGURATION

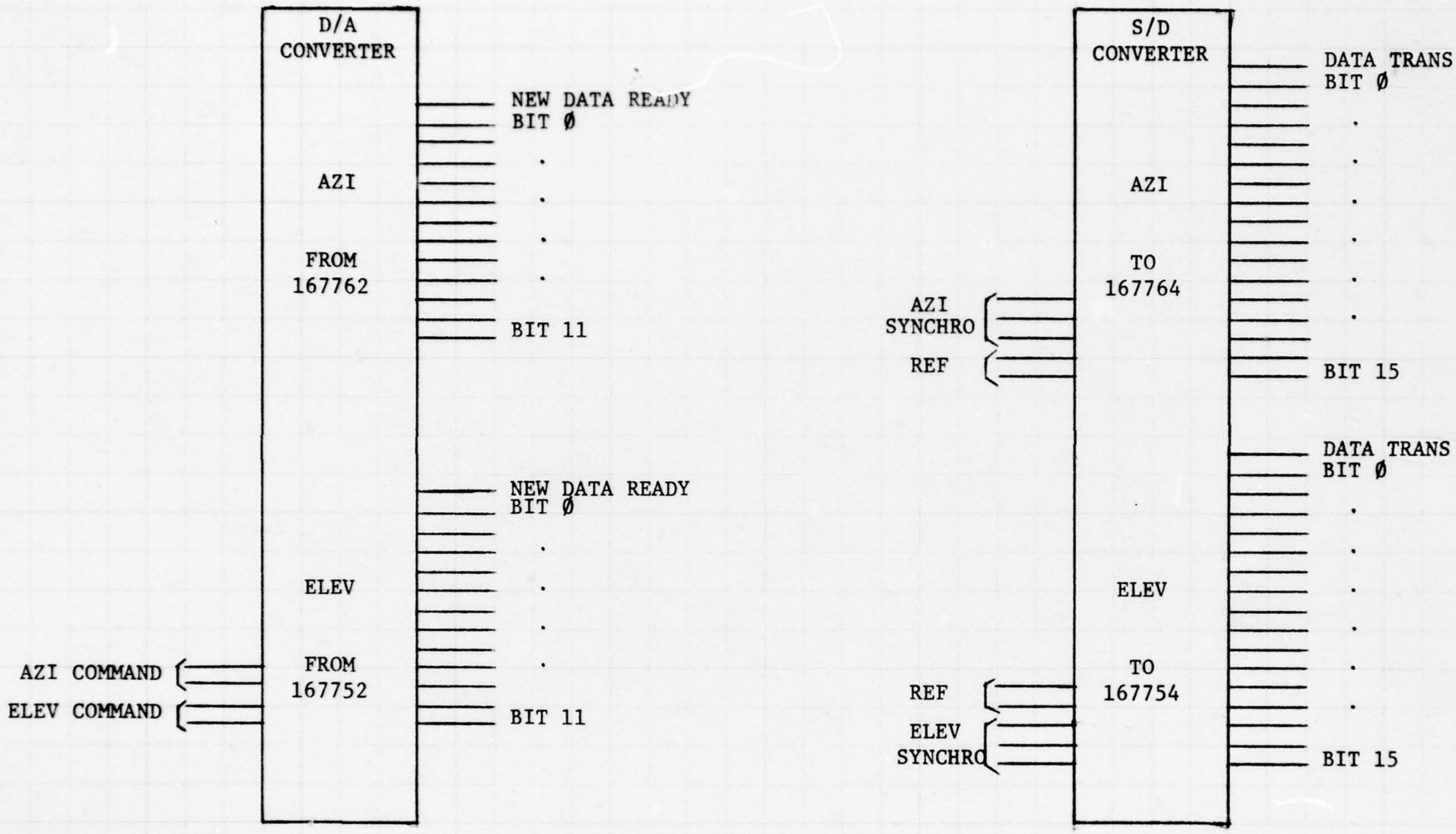


FIGURE 3

ANTENNA CONTROL INTERFACE

Ka-Band Rooftop Antenna Subsystem for Contract F33615-74-C-4012. It consists of a power supply, two digital to analog converters, and two synchro to digital converters. Its function is to convert digital pointing error commands as supplied by RAPP into analog azimuth and elevation error commands to the ten-foot antenna synchros. It also converts the analog azimuth and elevation voltages from the antenna servos into sixteen bit digital words, for use by RAPP in generating the aforementioned error commands. Figure 3 is a schematic representation of the ACIU.

2.2.2.1 11/45 to ACIU Interface. In the following discussion, all I/O is with respect to the ACIU.

2.2.2.2 ACIU Inputs. The desired azimuth and elevation error commands for the antenna servos are input to the ACIU from the 11/45 via two DR11-C general data interfaces and two D/A converters (see Figure 4). The Analog Devices DAC-12QM D/A converters used handle a twelve-bit (.00549325 degree LSB) data input word, provide a 10 volt output, and utilize an offset binary code. A thirteenth line strobes the digital data from the DR11-C into the DAC-12QM. Protocol is as follows: One of the RAPP subroutines transfers a twelve-bit digital error command to the output buffer of the appropriate DR11-C. DR11-C output buffer address 767752 is used for the elevation, while address 767762 is used for the azimuth. Once the program has loaded the error command to the output buffer, a positive going pulse (NEW DATA READY) is sent out to the appropriate D/A converter by the 11/45. The trailing edge of this pulse strobes the data from the DR11-C output buffer into the D/A converter.

The offset binary code used by the D/A converters requires an "all zeros" digital word for a +10 volt analog output. An "all ones" digital

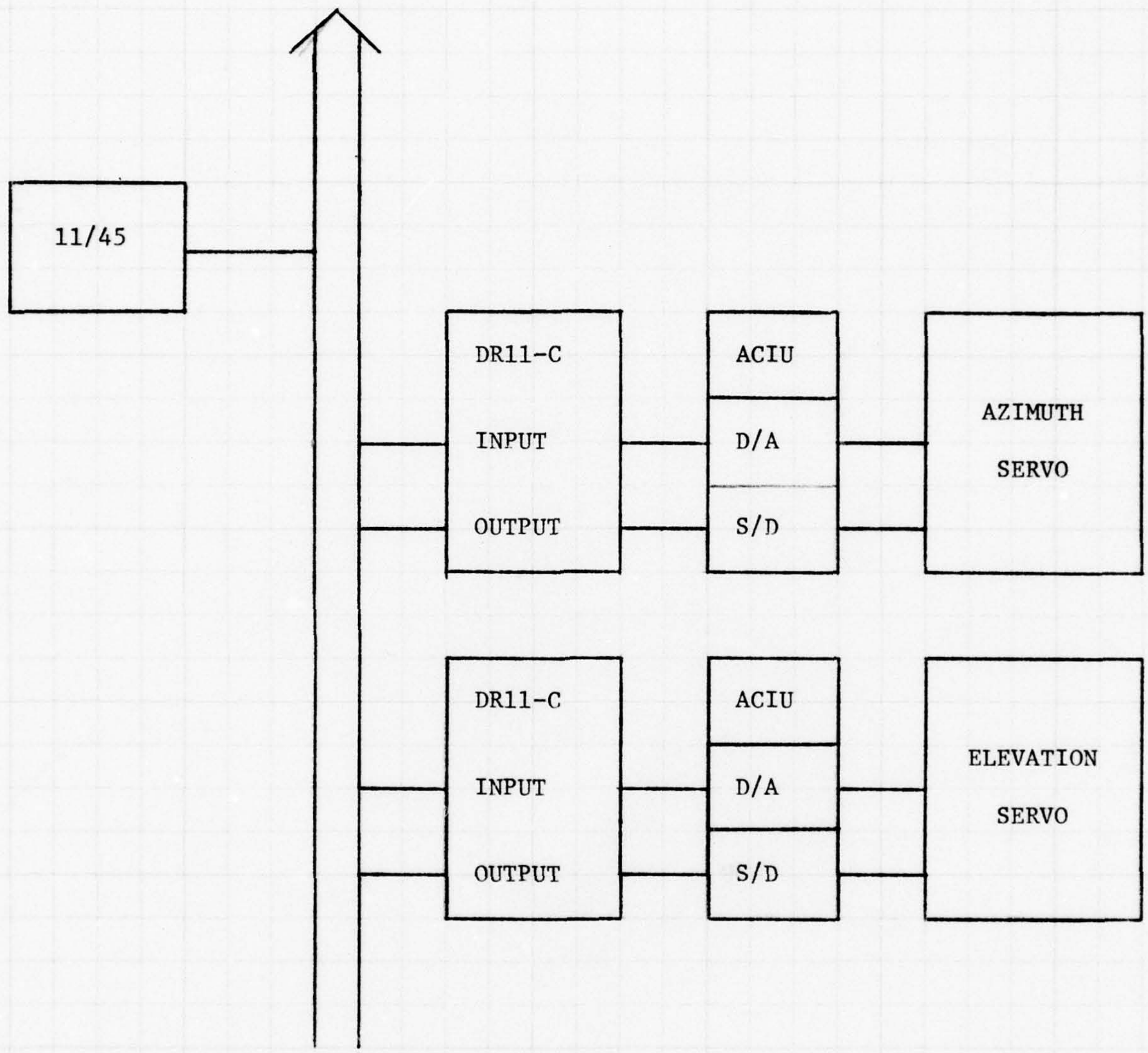

11/45
DR11-C
INPUT
OUTPUT
ACIU
D/A
S/D
AZIMUTH
SERVO
DR11-C
INPUT
OUTPUT
ACIU
D/A
S/D
ELEVATION
SERVO

FIGURE 4

word results in a -10 volt output. Octal 4000, midway between "all zeros" and "all ones", results in a zero analog output. In actual practice, full 10 volt output is not used. The software limits the digital error command word so that the magnitude of the peak analog voltage output by the D/A converter is 1.56 volts. This causes the antenna to move more slowly, thereby minimizing overshoot.

2.2.2.3 ACIU Outputs. The antenna azimuth and elevation angles are output via two Data Device Corporation A-Series Synchro to Digital Converters. In each case, a 16-bit two's complement digital word is provided to the 11/45, with the least significant bit being .00549325 of a degree. A seventeenth line is used as a HOLD for each S/D converter. When the 11/45 sets this line LOW, sampling by the S/D converter is inhibited. This allows the 11/45 to receive a complete and correct digital word. Once the word has been input, the 11/45 sets the HOLD line HIGH to allow sampling to continue.

2.2.3 Crosslink Ranging Receiver. The Crosslink Ranging Receiver, as supplied by Lincoln Laboratory, permits precise ranging to the satellite. Without going into detail, let it suffice that two 16-bit words, which represent the path delay to the satellite, are provided to the 11/45 from the Crosslink Receiver. DR11-C address 767774 is used to input the data. Specific details may be obtained from a number of memos and reports listed in the bibliography.

2.2.4 WWVB Receiver. Since the satellites' positions vary with the time of day, some means of obtaining an accurate time of day is necessary. To meet this need, a WWVB Receiver was purchased and then interfaced with the 11/45 under Contract F33615-75-C-1222. Time of day

($\underline{+}$30 msec) is supplied to RAPP as two 16-bit words via DR11-C address 767774. Specific details may be obtained from Technical Report AFAL-TR-76-145.

2.3 Software Introduction. The Rooftop Antenna Pointing Program (RAPP) runs on the PDP 11/45 computer under the RSX-11D Operating System. Ephemeris data, as supplied periodically by Lincoln Laboratory, contains the coefficients for a three-dimensional, ninth order curve fit of the orbits of LES 8 and LES 9. Using this data, with time as the independent variable, RAPP can point the ten-foot dish at either satellite.

Both satellites are in geosynchronous orbits, inclined approximately 23 degrees to the equator. This inclination causes the satellites to appear to move in "figure 8" patterns over a 24-hour period, with the center of the "8" occurring over the earth's equator. Therefore, it is important that RAPP be provided the correct date and exact time of day in order to accurately point to the desired satellite.

Of the fourteen RSX tasks which are a part of RAPP, seven are involved in actual antenna pointing, while the others are support programs. Four additional support programs operate under the Disk Operating System (DOS).

Unique aspects of RAPP are the use of a common memory block, the automatic installation of all antenna pointing tasks and support tasks, and the handling of ephemeris data. The implementation of these unique aspects, along with the RSX tasks and DOS programs, make up the software of the Rooftop Antenna Pointing Program.

2.4 Detailed Description of the Software:

2.4.1 Common Memory. In order to minimize the amount of memory required, a common area is set aside on the RSX disk for all constants and variables used by RAPP. Additional common area is also set aside for any future expansion of the program. FDUM1 and FDUM2 are available for floating point variables, while IDUM1 and IDUM2 are available for integers. The common area file is entitled BLODAT.FTN and is located under user number 1,1

on the system disk. FORTRAN programs reference it via COMMON statements, while Macro programs reference it via PSECT.

2.4.2 Installation of Tasks. Unlike DOS, all RSX tasks must be installed before being run. All antenna pointing support tasks may be automatically installed by invocation of the indirect install command "INS @INSTL". File INSTL.CMD consists of a list of tasks to be installed. This provides a time saving method of installing the tasks.

2.4.3 Ephemeris Data Handling:

2.4.3.1 Ephemeris Data Format. Satellite ephemeris data is stored on the system disk under user number 142,4 as are all user files. File DAVEM.DAT contains the ephemeris data for both LES 8 and LES 9. The data is periodically supplied to AFAL on a magnetic tape reel by Lincoln Laboratory. For any given day, there are four blocks of ephemeris consisting of A.M. (00:00-12:00z) and P.M. (12:00-24:00z) data for the two satellites. By dividing each day into two segments, more accurate look-angles can be computed. Figure 5 illustrates the format of data for the month of March.

Each block of ephemeris consists of thirty coefficients of the ninth order, three-dimensional curve fit of the satellite's position in earth-centered geocentric coordinates. Using the time of day as supplied by the WWVB receiver, RAPP computes look-angles to the desired satellite.

2.4.3.2 Ephemeris Data Transfer. Before the ephemeris data can be used by RAPP, it must be transferred from Lincoln Lab's magnetic tape onto the system disk. Lincoln generates the ephemeris data on an IBM 370 computer. The 370 transfers the data onto magnetic tape in blocks of 690 words. Unfortunately, the PDP 11/45 transfers data off the magnetic tape in blocks of only 256 words. This means that a normal

LES 8, 1 March A.M. coefficients

LES 8, 1 March P.M. coefficients

LES 8, 2 March A.M. coefficients

.

.

.

LES 8, 31 March P.M. coefficients

LES 9, 1 March A.M. coefficients

LES 9, 1 March P.M. coefficients

.

.

.

LES 9, 31 March P.M. coefficients

FIGURE 5. EPHEMERIS DATA FORMAT FOR MARCH

RSX file transfer under PIP or FLX would result in losing 434 words (690-256) from every block. Due to the short time schedule and familiarity with DOS software, it was decided to solve the problem under DOS rather than RSX. This results in a more complicated operator procedure, but it gets the job done. DOS program PAUL accomplishes this by use of the TRAN macro. See paragraph 2.4.5.7 for a description.

Once the ephemeris data has been transferred onto the DOS disk, it must be transferred to the RSX disk containing RAPP. This is accomplished by invoking FLX under RSX after mounting the DOS disk as a foreign disk. Specific details are included in paragraph 4.9, Section IV. The net result is the creation of RSX file DAVEM.DAT, which contains the LES 8/9 ephemeris data.

2.4.4 Antenna Pointing Operational Software. Antenna pointing is accomplished by seven interacting RSX tasks. RAPP, the master task, automatically runs three of the six remaining operational tasks. Two tasks associated with scanning the sky for a satellite are run by the operator. The seven operational tasks are summarized below.

RAPP - Run by the operator. Initializes all variables, inputs the data cards, reads ephemeris data off the disk, runs ASMINT, DOIT, and RTLOOP.

ASMINT - Run by RAPP. Initializes S/D and D/A converters used to control the ten-foot dish.

DOIT - Run and rescheduled every 200 milliseconds by RAPP. Calculates look-angles to the desired satellite using the ephemeris data. Issues commands via the D/A converters to point the ten-foot dish at the satellite. Monitors the antenna position via the S/D converters.

RTLOOP - Run and rescheduled every second by RAPP. Outputs pointing information (computed and measured azimuths and elevations, computed satellite range in kilometers and computed path delay in milliseconds). Runs DATAIL.

DATAIL - Run by RTLOOP. Looks up the appropriate ephemeris data whenever it is desired to change pointing to another satellite.

SCAN - Run by operator. Initializes conditions for starting a raster type scan of the sky. Run only when difficulty is experienced in locating the satellite. Runs SCANR.

SCANR - Run by SCAN. Performs a raster type scan by varying azimuth and elevation angles by ± 2.5 degrees about the nominal pointing.

2.4.4.1 Task RAPP Description. RAPP is the master program and, as such, performs the following functions:

A. Initialization of all common variables

B. Hardware initialization

C. Data card input via the card reader

D. Buffering of current ephemeris data from the disk

E. Conversion from geodetic to geocentric coordinates

F. Fixing of DOIT and RTLOOP into memory

G. Running of DOIT and RTLOOP

2.4.4.1.1 Common Variable Initialization. RAPP initializes all common variables prior to use.

2.4.4.1.2 Hardware Initialization. Hardware initialization is performed by task ASMINT. A detailed description of ASMINT can be found in paragraph 2.4.4.2, Task ASMINT Description.

2.4.4.1.3 Data Card Input and Location of Ephemeris. Before accurate pointing can begin, RAPP must know what day

of the year it is, which satellite is being used, weather information to correct for refraction error, and geographic location of the antenna. This information is contained on two data cards, read by subroutine DATAIO. The operator enters the day of year (1 = 1 January, 365 = 31 December, etc.), the satellite being used (8 or 9), the barometric pressure in inches of mercury, the temperature in degrees Fahrenheit, the relative humidity and the antenna's latitude and longitude.

Upon reading the data card, DATAIO corrects the latitude for the earth's oblateness, then computes the surface refractivity of the atmosphere. Finally DATAIO looks up the current day's ephemeris data using a direct disk access. Due to the format of the ephemeris data (see Figure 5), it may be necessary to read through almost all the ephemeris to locate the required day's data. Depending on the size of the ephemeris file, data look-up can take up to two minutes. Once the ephemeris has been located, it is stored in common memory.

2.4.4.1.4 Conversion from Geodetic to Geocentric Coordinates. Geographic locations are entered in standard geodetic coordinates. However, the ephemeris data, as supplied by Lincoln, contains the coefficients of a geocentric description of the satellite's position. Subroutine INITAL converts the antenna position from geodetic to geocentric coordinates.

2.4.4.1.5 Fixing and Running of DOIT and RTLOOP. The final action of RAPP is to FIX and RUN the two tasks which actually point the antenna. Both tasks are fixed in memory in order to avoid reading the tasks off the disk every time they are rescheduled. This saves both CPU time (which is now available for other users) and wear on the disk.

Once the tasks are fixed, RAPP runs them, unfixes itself (to free memory) and exits.

2.4.4.2 Task ASMINT Description. This task is run by RAPP to insure that the DR11 interfaces are initialized prior to pointing the antenna. The HOLD lines to the azimuth and elevation S/D converters are set HIGH to enable sampling. The REQUEST B line from the WWVB receiver is checked to insure that it is reset; if not, a dummy input is performed which should reset it, and it is checked again. Finally, "ZERO" pointing error commands are output to the antenna servos. Due to a small misalignment in the azimuth D/A converter, a 3775 offset binary code is output to the azimuth servos rather than a 4ØØØ.

2.4.4.3 Task DOIT Description. Task DOIT, as its name implies, does the actual pointing of the antenna. Using the ephemeris data located in common memory, it performs the following functions:

A. Inputs measured azimuth and elevation from the antenna's servos every 200 milliseconds.

B. Generates error commands to the antenna's servos every 200 milliseconds.

C. Computes a new look angle every five seconds.

2.4.4.3.1 Measured Look-Angle Input. Subroutine STATE is called by DOIT to input the current antenna look-angle. The procedure is to inhibit S/D converter sampling, input the look-angle, then re-enable sampling. Proper scaling is then accomplished prior to RETURN.

2.4.4.3.2 Error Command Generation. Subroutine PNT is called by DOIT to generate the look angle error command. Before an error command can be generated, there must be a reference system. Elevation is measured with zero degrees corresponding to the horizontal and 90 degrees corresponding to the vertical. Normally, azimuth is measured with zero degrees corresponding to true north, 90 degrees is

east, etc. However, for reference purposes only, PNT rotates the azimuth by 58 degrees towards the northwest, for reasons which will be explained in the new look angle calculation description. It should be understood that this rotation is done for the use of PNT only, and does not affect the antenna system's use of a zero azimuth corresponding to true north.

To perform the azimuth rotation, 59 degrees is added to both the computed and measured azimuths. Once this has been accomplished, both azimuth and elevation errors are calculated by subtracting the measured (actual) values from the calculated values. If the magnitude of either error exceeds ten degrees, a hard limiting effect is imposed, as illustrated in Figure 6. This prevents the antenna servos from being driven so hard that overshoot becomes a problem. If the magnitude of an error command is less than ten degrees, a unique method of generating the offset binary code is employed. Recalling that a zero error command corresponds to 2Ø48 (octal 4ØØØ), and noting that the LSB of the D/A converters corresponds to .03125, the decimal offset can be computed:

N = (zero error command) x LSB

= 2048 x .03125 - 64

Therefore, any error command can be converted to the offset binary code by adding 64 and dividing the sum by .03125. Once the offset binary error command has been calculated for both azimuth and elevation, PNT returns control to DOIT.

2.4.4.3.3 <u>New Look Angle Calculation and Exiting</u>. Since DOIT is rescheduled to run every 200 milliseconds, both STATE and PNT are called at the same rate. However, through the use of a counter, subroutine ANTENA is called once every five seconds. Using the ephemeris data, WWVB time of day (or computer time instead) and the surface refractivity

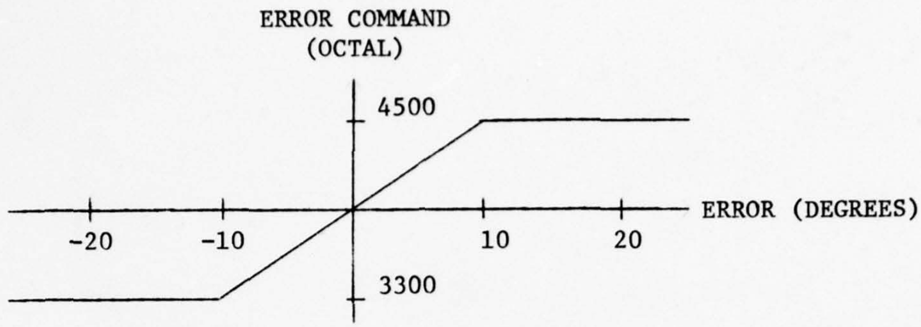


FIGURE 6. ERROR COMMAND LIMITING

ONES SEC (4 BITS)	HUNDREDS MSEC (4 BITS)	TENS MSEC (4 BITS)	ONES MSEC (4 BITS)

WORD 1

TENS HOURS (2 BITS)	ONES HOURS (4 BITS)	TENS MIN (3 BITS)	ONES MIN (4 BITS)	TENS SEC (3 BITS)

WORD 2

FIGURE 7. TIME OF DAY WORDS

constant, it calculates a corrected look angle.

Subroutine CTIME is called first by ANTENA to input the current time of day from WWVB and to check whether the A.M. or P.M. ephemeris data buffer is to be used. Before doing this, however, CTIME calls TOD.

Subroutine TOD inputs the current time of day from the WWVB receiver via DR-11C interface address 167774. TOD sends out a one-microsecond pulse via line CSRØ to clock out the first of the two 16-bit words containing the time of day. The WWVB receiver sends out a REQUEST B to the DR-11C when the data is ready. TOD then inputs the first word and repeats the process for the second word. The time of day is now contained in two 16-bit words as shown in Figure 7.

By performing various multiplications to effect shifts, TOD generates a floating point time of day expressed in seconds past midnight. TOD then returns control to CTIME.

Once WWVB time has been determined, it is compared with computer time. Since computer time is set from the WWVB display, the two should be close. However, upon implementation of the program, it was discovered that occasionally the wrong time of day would be loaded from the receiver. The "quick fix" was to compare computer time with the input from the WWVB receiver. If they differ by more than ten seconds, computer time is used.

Once the time of day has been properly determined, CTIME checks whether it is A.M. or P.M., loads the correct ephemeris buffer for use by ANTENA in the look angle calculation, and returns control to ANTENA.

With the time of day and correct ephemeris available, it is now possible to calculate the new look-angle. An iterative loop is used to compute the X, Y, and Z coordinates of the satellite position, using the following formula:

$$X_{SAT} = X_{10}t^9 + X_9t^8 + \dots X_2t + X_1$$
$$Y_{SAT} = Y_{10}t^9 + \dots Y_1$$
$$Z_{SAT} = Z_{10}t^9 + \dots Z_1$$

where X_1, X_2, ... Z_9, Z_{10} are the thirty coefficients contained in the ephemeris data.

Using the Avionics Lab position, the look-angle in earth centered geocentric coordinates is calculated next. However, with respect to the rooftop antenna system, the Avionics Lab is the center of the coordinate system. Therefore, it is necessary to rotate the look-angle into Avionics Lab coordinates. Then, taking the proper projections onto the local (Avionics Lab) horizontal and vertical planes, the azimuth and elevation angles to the satellite can be determined. Since the FORTRAN trigonometric functions only apply to one of four possible quadrants, the azimuth must be placed inside the proper quadrant (0°-90°, 90°-180°, 180°-270°, or 270°-360°).

Once the azimuth has been calculated, it must be checked to insure that the antenna is not pointed at the 15-story tower, which is a part of Building 620. Hazardous levels of RF radiation could endanger personnel or equipment located there. The "forbidden zone", then, is any azimuth between 276 and 302 degrees. DOIT subroutine PNT adds 58 degrees to a calculated azimuth in order to rotate the edge of the forbidden zone to the zero azimuth point. As mentioned previously, this rotation is for use by PNT only and does not affect the antenna system's reference. Since the envelope of azimuths used in pointing at the satellites is between 90 and 270 degrees, the forbidden zone will have no effect during LES 8/9 testing.

Final azimuth and elevation angles are obtained by adding any bias terms (see task RTLOOP description for discussion of biases), and by correcting for refraction.

Subroutine REFCT, provided by Lincoln Laboratory, corrects for refraction. By an iterative process of calling subroutine DELL and adding the correction, the calculated elevation angle is corrected for refraction.

Subroutine DELL (also provided by Lincoln Laboratory) uses the surface refractivity constant available from DOIT subroutine DATAIO to provide refraction correction terms to REFCT.

Upon correction of refraction, ANTENA returns control to DOIT, which then exits. The corrected look angle is now ready for use.

2.4.4.4 Task RTLOOP Description. RTLOOP polls the Rooftop VTØ5 display terminal every five seconds to determine if:

A. It is desired to obtain the latest pointing information.

B. It is desired to insert an azimuth or elevation bias term.

C. It is desired to change pointing to the other satellite.

2.4.4.4.1 Obtaining Pointing Information. Pointing information is obtained by typing "INFO" on the Rooftop VTØ5 keyboard. Upon detecting the "I" in "INFO", subroutine AEOUT will display the current pointing information, as illustrated in Figure 8. Azimuth and elevation are displayed in addition to ranging information. Range to the satellite is expressed in three different ways: decimal kilometers, offset octal kilometers, and offset octal milliseconds. The two octal formats are generated for use by personnel operating the Spread Spectrum Modem Processor located in the Rooftop Facility. Offset octal range in kilometers is determined by

$$R_{OK} = R_{DK} - 10000.$$

where R_{OK} = Octal range in kilometers

R_{DK} = Decimal range in kilometers

Offset octal range in milliseconds is determined by

$$R_{OM} = \frac{R_{DM}}{.3} - 131000.$$

where R_{OM} = Octal range in milliseconds

R_{DM} = Decimal range in milliseconds

Decimal azimuth and elevation error terms are displayed to indicate any error between the calculated and measured pointing angles. If the error becomes unacceptably large, the D/A converters in the Antenna Control Interface Unit should be adjusted.

```
I
 AZIMUTH=  200.762
 ELEVATION=    13.673
 RANGE=  40239.9 ( 73037)
 PATH DELAY= (  6074)
 AZIMUTH ERROR=  200.763
 ELEVATION ERROR=  163.029
```

FIGURE 8. RANGING INFORMATION PRINTOUT

2.4.4.4.2 Inserting Bias Terms. Several factors can cause less than exact pointing. These include slight antenna misalignment and poor orbit fitting in generating the ephemeris data. Both can be offset by inserting bias terms on the Rooftop VTØ5 keyboard. Azimuth bias is inserted by typing "A", followed by a five-digit bias term. Elevation bias can likewise be entered by typing "E", then the bias term. Subroutine CONVER, called by AEOUT will convert the bias term into the proper form by use in pointing. It does this by scanning the five digit input field for a decimal point and scaling the exponent accordingly. If the bias term exceeds 360 degrees for azimuth or 90 degrees for elevation, an error message will be output on the VTØ5 display.

2.4.4.4.3 Changing Satellites and Exiting. Satellite pointing is changed by typing "L", then "8" or "9" as the case may be. Subroutine AEOUT detects the "L". If any number other than 8 or 9 is entered, an error message will result. Task DATAIL is run to look up the appropriate ephemeris, as discussed in paragraph 2.4.4.5.

Whether or not DATAIL is run, RTLOOP then exits, completing one cycle of antenna pointing. RTLOOP is rescheduled to run in one second. A counter causes the VTØ5 to be polled once every five times RTLOOP is run.

2.4.4.5 Task DATAIL Description. When DATAIL is run by RTLOOP, three things happen:

A. Tasks DOIT and RTLOOP are cancelled

B. New ephemeris is looked up

C. Tasks DOIT and RTLOOP are rescheduled at their respective intervals

2.4.4.5.1 Cancelling of Tasks. DOIT and RTLOOP are cancelled by DATAIL to inhibit antenna pointing, and to provide maximum CPU time for DATAIL to look up the ephemeris.

2.4.4.5.2 Ephemeris Look-Up. Since DATAIL is modified from the RAPP subroutine DATAIO, it uses the same direct disk access method to look up the ephemeris data.

2.4.4.5.3 Rescheduling of Tasks and Exiting. Once the ephemeris has been located and buffered, DOIT and RTLOOP are rescheduled at their respective intervals. This action allows pointing at the new satellite to begin. DATAIL then exits.

2.4.4.6 Task SCAN Description. Task SCAN, in conjunction with task SCANR, provides the capability to perform a raster type search for the satellite. It is only used when the satellite cannot be located by RAPP. It can also be used in making antenna pattern measurements. SCAN is run manually from the DECWRITER, while SCANR is run automatically by SCAN. The functions of SCAN are:

A. Cancellation of tasks DOIT and RTLOOP

B. Running of SCANR and Implementation of the Scan

C. Cancellation of DOIT and RTLOOP

2.4.4.6.1 Cancellation of DOIT and RTLOOP. Tasks DOIT and RTLOOP are cancelled to prevent interference while SCAN is trying to perform the raster scan.

2.4.4.6.2 Running of SCANR and Implementation of the Scan. SCAN runs task SCANR and reschedules it to run five times a second. SCANR performs a function similar to task DOIT, in that it utilizes the same subroutines as DOIT. Upon being run, SCAN prints "UNABLE TO INITIALIZE SCAN" if for some reason it is unable to move the antenna to within one tenth of a degree of its desired starting location.

Once the antenna is in its initial position, SCAN passes successively increasing biases to SCANR in two-tenth-of-a-degree steps. Each time SCANR

runs (five times per second), it moves the antenna an additional two-tenths of a degree in azimuth. Once five degrees of azimuth has been swept through, SCAN subtracts two-tenths of a degree from the elevation. SCANR then uses this bias to drop the elevation. SCAN then passes increasing, negative biases to SCANR in two-tenth-of-a-degree steps, causing an azimuth sweep in the other direction. The process continues until a five-degree square of the sky has been scanned, as illustrated in Figure 9.

Figure 9 is a simple illustration, because during the two minutes that the scan takes, the center of the raster square is moving along the path of the satellite. In other words, each azimuth and elevation bias is added (or subtracted) from the current exact look angle to the satellite, rather than to the angle which existed when the scan began.

Coordination between SCAN and SCANR is important, to insure that SCAN passes the correct bias to SCANR at the proper time. For this purpose, a flag word is set each time SCANR runs. By testing this flag, SCAN either waits for SCANR to run, or passes the bias to SCANR. As an aid to the operator, SCAN prints "END OF ONE AZIMUTH SCAN" on the Rooftop VTØ5 every time an azimuth scan is completed. It also prints the current azimuth and elevation every second on the Rooftop VTØ5.

2.4.4.6.3 Cancellation of SCANR, Rescheduling of Tasks, and Exiting. Upon completion of the raster scan, SCAN prints "SCAN COMPLETE" on the Rooftop VTØ5. Task SCANR is then cancelled. Task DOIT is then rescheduled to run every 12 ticks, while task RTLOOP is rescheduled to run every second. This resumes the antenna pointing. Prior to exiting, SCAN zeros the azimuth and elevation biases.

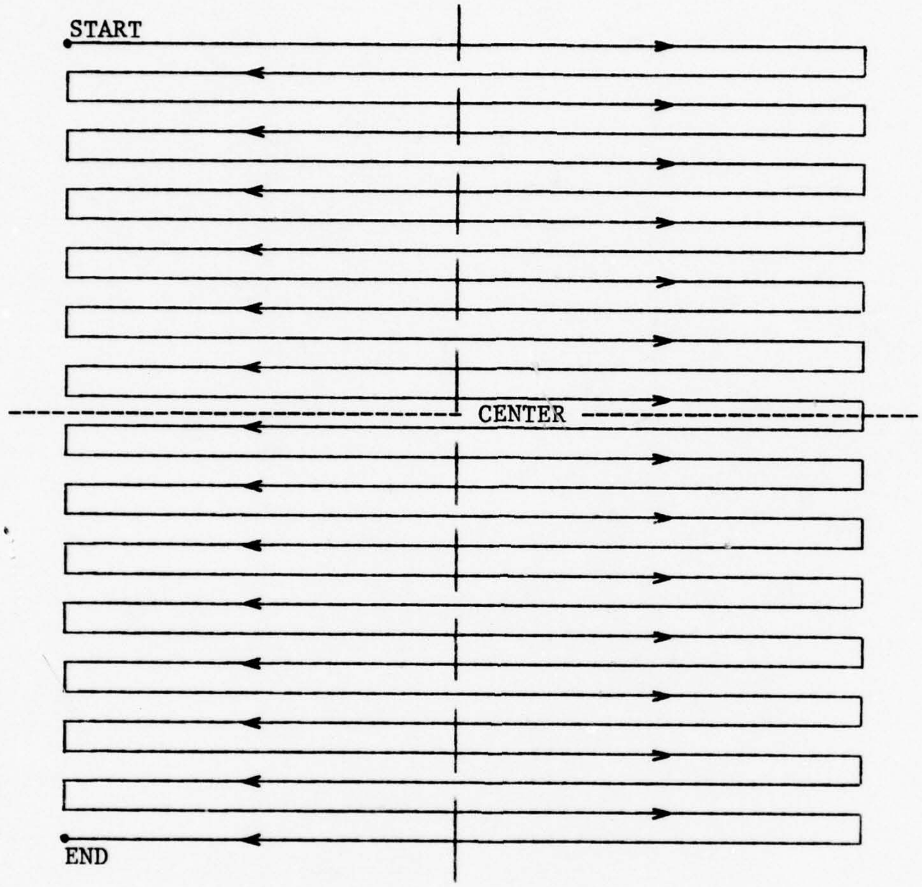


FIGURE 9. RASTER SCAN

2.4.4.7 Task SCANR Description. Task SCANR is identical to task DOIT, except for three changes. While DOIT calls subroutine ANTENA every five seconds, SCANR calls a modified ANTENA subroutine (PANTEN) at a five-time-per-second rate. The modification consists of the addition of the azimuth and elevation bias terms, added by use of a COMMON block (BBIAS). By calling PANTEN every time SCANR runs (five times per second), the raster scan is able to follow the motion of the satellite during the two-minute scan. The third change is the use of the flag which aids in the coordination between SCAN and SCANR.

Except for these three changes, the DOIT description may be used as a description for SCANR.

2.4.4.8 Operational Software Conclusion. This completes the description of the operational software used in pointing the Rooftop Antenna. Complete program listings are included as part of Section III. By following the descriptions of this section, along with the information in Section III, it should be possible to augment or modify any of the operational software.

2.4.5 Antenna Pointing Support Software. Six RSX tasks and four DOS programs make up the support software. RSX tasks HAWAII and ALASKA print look angles to either LES 8 or LES 9 from any point on the earth's surface. RSX task WRITER prints the current look angle and the calculated look angle when the Rooftop Terminal is in the Active Track mode. This makes it possible to observe any errors between the computed and actual look angles. RSX task DATA lists the ephemeris data on the line printer, for use by TRW personnel in entering the ephemeris by hand into the K-Band Modem located in the Rooftop. RSX task XLINK initializes a transfer of pointing data from the antenna, the Crosslink Ranging Receiver, and the WWVB receiver onto magnetic tape. Task INPUT is called by XLINK to perform the actual data input and buffering prior to transfer onto tape.

DOS programs RCA and TRW punch a paper tape of the ephemeris data for other users. DOS program PAUL transfers the ephemeris data from magnetic tape onto the disk. DOS program SATSPT calculates look angles from any point on the earth to any satellite position, given the satellite latitude, longitude, and altitude.

2.4.5.1 Task HAWAII Description. Task HAWAII is so named because it was originally developed for predicting look angles for a flight to Hawaii. Task HAWAII was developed by making various changes to RAPP.

HAWAII predicts the hourly look-angles to either satellite, from any point on the earth. Allowances were not made for the aircraft's altitude since even 50,000 feet is small compared to the satellites' altitude. A short description follows of the various changes:

A. Changes to RAPP

B. Changes to the initialization routine

C. Changes to the ephemeris look-up routine

D. Changes to the look-angle calculation routine

E. Changes to the time-of-day routine

2.4.5.1.1 Changes to RAPP. The major changes to RAPP involved providing for repetitive ephemeris look-up and printing of the look angles.

Since a typical test flight spans several days, the look-up of several days' ephemerides must be allowed for. RAPP is set up to look up one day's ephemeris at a time. If RAPP's ephemeris data look-up subroutine were called more than once, an error message would result due to the CALL ASSIGN and DEFINE FILE statements being invoked more than once. Therefore, these two statements are moved into HAWAII, the main program. In this way, repetitive ephemeris look-up is allowed for.

Once a look-angle for a given hour in a given day is computed, it is printed on the line printer, along with the hour, satellite number (8 or 9), and the day-of-year. The time is incremented by one hour, and a new look-angle is computed and printed. The process continues for the 24 hours of the day, and as for as many days as requested. Finally, since antenna pointing is not required, tasks DOIT and RTLOOP are not run.

2.4.5.1.2 Changes to the Initialization Routine. Since HAWAII does not involve antenna pointing, none of the hardware needs to be initialized. Therefore, HINT, the modification of INITAL, does not run task ASMINT.

2.4.5.1.3 Changes to the Ephemeris Look-Up Routine. As mentioned previously, subroutine HDATA, the modification of DATAIO, does not invoke CALL ASSIGN or DEFINE FILE. In order to make data card preparation easier, only one data card is used as opposed to the two

RAPP requires. It includes the day of the year, the observer's position, and satellite number (8 or 9). Nominal weather data (72 degrees F, 30 inches of mercury, and 50% relative humidity) is used for refraction correction.

In addition, a pointer is provided to facilitate rapid ephemeris look-up. This allows HDATA to "remember" where it was in the ephemeris data, so that it is unnecessary to start at the beginning of the ephemeris each time HDATA is called. This feature is especially useful when working with LES 9.

Finally, the printing of the ephemeris on the line printer, which DATAIO does, is not done by HDATA. Obviously, this would interfere with the printing of the look-angles, and is unnecessary.

2.4.5.1.4 Changes to the Look-Angle Calculation Routine. Once the ephemeris data has been located, the calculation of the look angle is undertaken. Subroutine ANTENA was modified to remove the code which prevents the antenna from being pointed at the twin towers. The new subroutine, HANTEN, can therefore generate any azimuth from Ø to 36Ø degrees.

2.4.5.1.5 Changes to Time-of-Day Routine. Since it is unnecessary to input the time-of-day from the WWVB receiver, the calling of that particular subroutine was deleted in the new subroutine HTIME. As mentioned previously, the time-of-day is controlled by the main program, HAWAII.

2.4.5.1.6 Conclusion. Task HAWAII, being a modification of RAPP, has been used successfully in predicting look angles to LES 8 and 9 for flights in both the Northern and Southern Hemispheres, as well as the Western and Eastern Hemispheres and the North Pole.

2.4.5.2 Task ALASKA Description. Task ALASKA is so named because it was originally developed for predicting look angles for a flight to Alaska. Based on the flight plan, ALASKA prints the look angle to the

satellite at a desired time of day, plus the angle one, two, and three hours later. If takeoff is delayed, the ALASKA data is useful for altering the test plan, thus maximizing the probability of a successful test. Task ALASKA was developed by making a modification of HAWAII.

2.4.5.2.1 Modification of HAWAII. Task HAWAII was modified by altering the code in generating the time of day, and by providing code which queries the operator as to the time it is anticipated each point on the flight path will be reached. Whatever time of day is input, ALASKA will compute the look angle at that time, then increment the time by one, two, and three hours and calculate the look angle for each time. Except for this change to task HAWAII, ALASKA uses the same subroutines as HAWAII does.

2.4.5.3 Task WRITER Description. Task WRITER can be used whenever the antenna is being driven by the Rooftop Terminal in the Active Track mode. It is used to observe the differences between the computed and actual look angles. It operates by "stealing" the time-of-day from RAPP, the computed and measured look angle, and the azimuth and elevation error. It then simply prints this information on the line printer. By using the reschedule option of the RUN command, WRITER can be automatically rescheduled at any desired interval.

2.4.5.4 Task DATA Description. Task DATA is used to list the ephemeris data from the disk onto the line printer. This is done to provide the TRW personnel with a copy of the ephemeris data to use when they key in the ephemeris into the K-Band Modem computer. Task DATA uses double precision in order to print each thirteen-digit coefficient that the K-Band Modem computer requires.

2.4.5.5 Task XLINK Description. Task XLINK performs the function of recording all the parameters that Lincoln Laboratory requires to perform orbit fitting for each satellite. The resulting data is sent to Lincoln Laboratory to be used in generating new ephemeris data, one to two months in advance.

2.4.5.5.1 Data Recorded by XLINK. In order for the data to be used by Lincoln's orbit fitting program, the data must be stored on magnetic tape in a specific order. Figure 10 illustrates the format for each block of data. The data is collected and recorded once every four seconds, during the first ten minutes of every hour.

2.4.5.5.2 Implementation of XLINK. Task XLINK is run only when the Rooftop terminal is actively tracking a satellite. Scheduled to run every four seconds, XLINK first checks the time of day via the WWVB receiver. If the time is within the first ten minutes after the hour, XLINK sets an INPUT RUN flag, runs task INPUT, and exits. Task INPUT collects and stores the required pointing data, and SUSPENDS itself. With the INPUT RUN flag set, task INPUT will RESUME when commanded to do so by XLINK. Once 150 sets of data have been collected and recorded (150 sets x 4 times per second = 10 minutes), XLINK resets the INPUT RUN flag. With this flag reset, XLINK will resume checking of the time of day, until the next hour. If it is desired to complete the data collection and recording, BIT ZERO of the 11/45 Switch Register should be set. This causes an end-of-file to be written on the mag tape. The tape then rewinds itself automatically, completing the action.

2.4.5.6 Task INPUT Description. Task INPUT does the actual collection of the data. Time-of-day is input from the WWVB receiver and formatted prior to being buffered for output. Azimuth and elevation

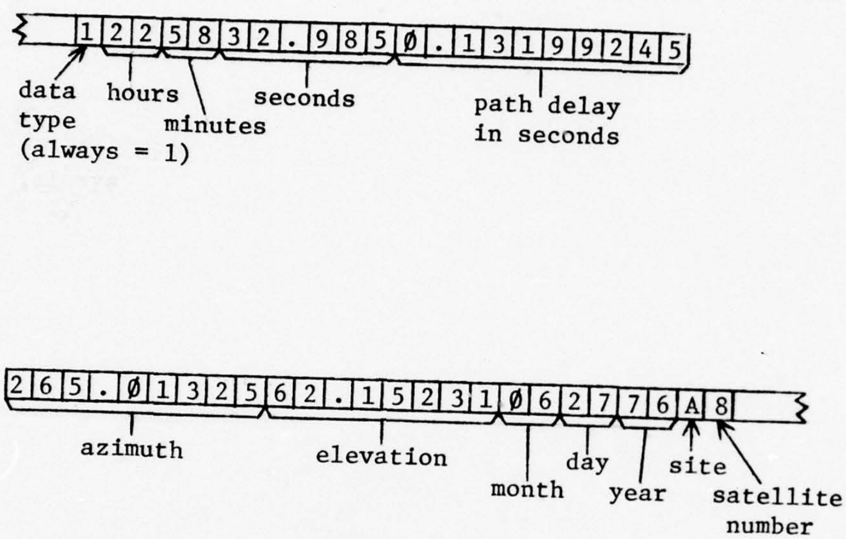

1 2 2 5 8 3 2 . 9 8 5 Ø . 1 3 1 9 9 2 4 5
data type (always = 1)
hours
minutes
seconds
path delay in seconds
2 6 5 . Ø 1 3 2 5 6 2 . 1 5 2 3 1 Ø 6 2 7 7 6 A 8
azimuth
elevation
month
day
year
site
satellite number

FIGURE 10.

are recorded, and the path delay is input from the Crosslink Ranging Box located in the Rooftop. Task INPUT also buffers the constant type of data - data type, month, day, year, site, and satellite number. Once 150 blocks of data are buffered, they are written on magnetic tape.

Once the ten-minute data is written on tape, INPUT suspends itself, until resumed by XLINK on the hour. If it is desired to close out the data file on mag tape, bit Ø on the 11/45 switch register must be set. This causes INPUT to exit, which closes the tape file.

2.4.5.7 Program PAUL Description. Program PAUL runs under the Disk Operating System (DOS) on the PDP 11/45. It is used to transfer ephemeris data from magnetic tape onto the disk.

2.4.5.7.1 Implementation of PAUL. The IBM 370 located at Lincoln Laboratory, which is used to generate the magnetic tape, writes the ephemeris data in blocks of 690 words.

The PDP 11/45 is set up to transfer data in blocks of 256 words. This means that 434 (690-256) words would be lost from every block, making the transferred ephemeris useless. As a solution to this problem, Program PAUL was written in the PDP 11 Assembly Language. It sets up a file on the disk entitled DAVEM.DAT, and performs a TRAN 690 times. The TRAN takes a word from the magnetic tape and transfers it into the disk file DAVEM.DAT. Each block is transferred in this way until two end-of-files have been encountered (one each for LES 8 data and LES 9 data).

2.4.5.8 Program SATSPT Description. Program SATSPT runs under DOS Version 8 on the 11/20. Version 8 is used because it is the only version with a Plotter driver routine. SATSPT is a great circle navigation program, computing look angles from any point on the earth to any satellite. SATSPT is a modified version of a program originally written for a CDC 6600 computer by Lt Joseph Knapick of the 4950th Test Wing here at Wright-Patterson. The modifications consist of adding library functions not found on the PDP 11, adding a plotting capability, and providing a capability of world-wide observer positions. Input requirements are the observers position in latitude and longitude, and the altitude, latitude, and longitude of the satellite.

Finally, iso-elevation flight paths can also be plotted, knowing the required satellite positions.

2.4.5.8.1 Implementation of SATSPT. With slight modification, the following discussion is adapted from the Program Description and Operating Instructions for "SATSPT: An Aircraft/Satellite Relationship Program with Iso-Elevation Line Calculation Options", prepared by Lt Joseph F. Knapick, Jr. on 28 October 1975.

2.4.5.8.1.1 Output Options. Several input options provide alternate methods of position data read-in. Positions may be input in decimal format, or degrees and minutes format. Additionally, aircraft position may be expressed as a range and bearing from a VORTAC station or other navigational aid. The four output options are:

Option Ø: Calculation of elevation angle and true azimuth to the satellite, given aircraft position, satellite sub-point and satellite altitude. Output positions are in decimal format.

Option 1: Same as option Ø, except output positions are in degrees and minutes format.

Option 2: Calculation of aircraft positions which lie on a line of constant elevation angle specified in the input data. The result is an iso-elevation angle line-of-position. The calculation is executed over a range of aircraft longitudes specified as input data. Zero, one, or two latitude solutions are possible for any given longitude. The unknown latitude(s) is (are) determined for each longitude; then the calculations proceed per output option 1. The program results display all valid solutions, noting if more than one solution exists. (Exception: when the satellite subpoint lies on the equator, only the northern hemisphere solution is printed.) Since these solutions are symmetrical about the equator, the second solution may be obtained by reversing the sign of the displayed latitude and reflecting the true azimuth about the equator.

Option 3: Repeats the process described under option 2 for a range of elevation angles specified in the data cards. An iso-elevation map results. The output data may be plotted immediately on the CALCOMP plotter located in the CSEL lab.

For each option, most input data and results of supplemental calculations are also displayed:

1. Observation time
2. Aircraft true heading
3. Aircraft pitch
4. Slant range (nautical miles)
5. Elevation angle corrected for aircraft pitch (options Ø and 1 only)
6. Relative bearing (azimuth to satellite with respect to the aircraft nose)
7. VORTAC station latitude and longitude (when applicable)
8. Magnetic bearing and distance (nautical miles) to VORTAC
9. Magnetic variation at aircraft/VORTAC
10. Distance from aircraft position to WPAFB (nautical miles)
11. Great Circle distance between the aircraft and the satellite subpoint
12. Remarks applicable to the calculation (input data and/or program generated)

Error detection tests throughout the program terminate execution if a serious input error is discovered. An appropriate diagnostic statement is printed prior to program termination.

2.4.5.8.1.2 Addition of Library Functions. While ARCSIN, ARCCOS and TAN are standard features of the CDC 6600 (for which SATSPT was originally written), they are not included

on the PDP 11. As a result, both functions were implemented using the arc-tangent function, which is a part of the PDP 11's library. To obtain the arcsine in degrees from the arctangent,

$$\sin^{-1}(X) = 57.2957795 \quad \tan^{-1}\left(\frac{X}{\sqrt{1-X^2}}\right)$$

To obtain the arccosine in degrees,

$$\cos^{-1}(X) = 57.2957795 \quad \tan^{-1}\left(\frac{\sqrt{1-X^2}}{X}\right)$$

Likewise, the tangent is determined simply by dividing the sine by the cosine.

2.4.5.8.1.3 Addition of Plotting Capability. When desired (as explained in Section IV), a plot can be obtained of azimuth or elevation vs. observer position, iso-elevation flight paths, or iso-azimuth flight paths. Each plot is generated on the CALCOMP plotter and is scaled to overlay on a map of the world. The plotting capability has proven extremely useful in planning and/or verifying flight paths to be flown during a particular test.

2.4.5.8.1.4 Addition of Subroutine AZI. The original SATSPT was written for use in the northern half of the Western Hemisphere only. As a result, erroneous azimuths were generated for flights planned in any other hemisphere. Subroutine AZI was written to take into account some of the peculiarities of Great Circle navigation. For instance, one might think that an initial bearing directed due west of Wright-Patterson would eventually pass through China. On the contrary, it would pass through central Australia. Such peculiarities have led to the inclusion of code to AZI which, for a given longitude, determines the magnitude of the latitude at which the satellite would lie on an east-west Great Circle path. To determine the sign of the latitude (north or south),

AZI "plugs in" the northern latitude into the azimuth calculation.

AZI then "plugs in" the southern latitude into the azimuth calculation. Whichever latitude has an azimuth closer to $\pm 90^{o}$ or $\pm 270^{o}$ is assigned as the east-west reference latitude. Once the east-west reference is determined, AZI can correctly determine the quadrant in which the azimuth to the satellite lies.

2.4.5.9 Program RCA Description. Program RCA runs under DOS on the PDP 11/20 computer located in the CSEL lab. RCA punches the ephemeris data on paper tape, for use by an IBM Four Pi computer located on the C135 test aircraft. The IBM computer is used in pointing the aircraft three-foot dish at the satellite. Program RCA reads the ephemeris off the disk, formats into BAUDOT, then punches the reformatted data onto paper tape. A special paper tape driver was written to punch the tape.

2.4.5.9.1 Implementation of RCA. Program RCA reads the day-of-year from the 11/20 console. It then looks up the corresponding ephemeris for both LES 8 and LES 9, and after some preliminary reformatting the ephemeris is spooled back onto the disk under file RCA.LST. RCA then reads from file RCA.LST and formats the ephemeris into BAUDOT. An assembler paper tape driver then punches it on paper tape. Upon completion, file RCA.LST can be listed on the line printer.

2.4.5.9.2 Punching Paper Tape. Assembly language program IO drives the high speed paper tape punch. Receiving characters from RCA, IO punches the tape in the desired format by addressing the output buffer of the high speed paper tape punch interface.

2.4.5.10 Program TRW Description. Program TRW runs under DOS on the PDP 11/20 computer located in the CSEL lab. TRW makes the ephemeris data available for the ROLM Rugged Nova computer used in the

Rooftop K-Band Modem. TRW reads the ephemeris data off the disk and punches the data via an assembly subroutine onto paper tape. Like Program RCA, TRW reads the day-of-year from the 11/20 console and uses subroutine IO to punch the data on paper tape. Unlike RCA, TRW does no reformatting of the ephemeris - the ROLM computer uses it unaltered. As a result, no intermediate data is spooled onto the disk.

2.5 Building Tasks and Linking Programs. All of the tasks used in the Rooftop Antenna Pointing Program were built using the indirect command feature of RSX. Each command sequence is located on the disk under the .CMD extension. As an example, RTLOOP.CMD contains the command sequence to build task RTLOOP. When building RTLOOP or INPUT under TASK BUILDER, dummy FORTRAN main programs RTL or INP call the RTLOOP or INPUT assembler programs. This makes it possible for the assembler programs to write data on any peripheral using the FORTRAN WRITE statement.

DOS does not feature the indirect command feature so each program must be linked manually. The command sequence used in linking each of the four DOS programs appears in Figure 11.

2.6 Future Modifications of RAPP. Future applications of RAPP may include the requirement to accurately point at some other satellite(s) operating in the Ka band, Q band or beyond. As long as the ephemeris data is in the same ninth order polynomial form, no changes would need to be made to RAPP. However, it is anticipated that future satellites would use the ADBARV format of the ephemeris data. In that event, only subroutines DATAIO and ANTENA would have to be modified and their associated tasks rebuilt to accommodate the new format.

It may be also desirable to make the above modifications if the ADBARV ephemeris could be obtained for the several other satellites now

```
#PAUL<PAUL/E

#RCA<RCA,IO,FTNLIB/L/E

#TRW<TRW,IO,FTNLIB/L/E

#SATSPT<SATSPT,PLMODS/CC,PLTLIB/L,FTNLIB/L/E
```

FIGURE 11. LINKING DOS PROGRAMS

being tested by the Satellite Communications Group. This would provide the AFAL with an in-house capability to predict look angles in planning flight paths for testing.

Another change which could be made would be to enlarge the ephemeris data buffers to accommodate several days' ephemeris for two or more satellites. In its present form, RAPP only buffers twenty-four hours of ephemeris for one satellite. Switching satellites now requires on the order of two minutes to accomplish. While not a long time, it causes an unnecessary delay. Furthermore, RAPP can only run unattended for twenty-four hours at a time, since that is all the ephemeris data it buffers. By expanding the data buffer to include several days' ephemeris and providing for a change of day at 2400Z, RAPP can be improved. Modifications would involve task BLODAT, and subroutines DATAIO and CTIME. In addition, tasks RAPP and RTLOOP would have to be rebuilt.

A third change would involve modifying RAPP subroutine DATAIO to input only one rather than two data cards. While originally there was a need for more input data, the present program only requires the day number, satellite number, and weather information.

A fourth change would involve incorporating permanent azimuth and elevation biases into RAPP, once sufficient data has been collected via task WRITER. RAPP would then automatically remove any biases inherent in the antenna pointing system.

Finally, RAPP could be adapted to run under the RSX-11M operating system. Since "M" has been introduced, the day is approaching that "D" will no longer be supported by DEC. Conversion to "M" should be simply a matter of re-compiling and rebuilding the tasks. At this time, a routine should be written which will transfer the ephemeris data directly onto the RSX disk from magnetic tape. This would replace DOS program PAUL.

SECTION III. PROGRAM LISTINGS

TASK RAPP

```
C          ROOFTOP ANTENNA POINTING PROGRAM (RAPP)
C
C          THIS TASK IS INSTALLED UNDER THE RSX-11D OPERATING SYSTEM.
C          RAPP GENERATES THE COMMANDS NECESSARY TO POINT THE 10 FT
C          DISH ANTENNA LOCATED ON TOP OF THE
C          U.S. AIR FORCE AVIONICS LABORATORY  BUILDING 620
C
C          INPUT DATA HAS UNITS OF DEGREES, METERS, AND SECONDS
C
      COMMON /TT/TMID,TETIM,TF,TI,TETI3,DT
      COMMON /AL/ALLAT,ALLNG,ALRAD,RAD,SLAT,SLNG,CLAT,CLNG
      COMMON /C/XC(12),YC(12),ZC(12),XCA(12),YCA(12),ZCA(12)
      COMMON /C/XCB(12),YCB(12),ZCB(12),KK,IVAR
      COMMON /COORD/XSAT,YSAT,ZSAT,XAL,YAL,ZAL,XX,YY,ZZ,XXA,YYA,ZZA
      COMMON /ANT/AZI,ELEV,AZIIN,ELEIN,SDBIT,DABIT,ERA,ERE
      COMMON /FL/K,L,M
      COMMON /BIAS/AZBIAS,ELBIAS,TIBIAS
      COMMON /SATC/XSATC,YSATC,ZSATC,DTC
      COMMON /DOTI/XDOTI,YDOTI,ZDOTI,XDOT,YDOT
      COMMON /TDDOP/FREQ,DOPLER,TDLAY
      COMMON /PTHN/PP,TC,TK,PW,AN
      COMMON /ANTC/IOUNT,INTFLG
      COMMON /IDUM1/LES,N,IEOF,IAMFLG,IPMFLG,I6,IERFLG,I8,I9,I10
      DATA RTLOOP/6RRTLOOP/
      DATA RAPP,STATE1,PNT1,DOIT/6RRAPP  ,6RSTATE1,6RPNT1  ,6RDOIT  /
C          PROGRAM NAME AND INSTRUCTIONS
      WRITE(6,100)
      WRITE(6,101)
C          INITIALIZATION ROUTINE
      IEOF=0
      IAMFLG=0
      IPMFLG=0
      IERFLG=0
      TMID=0.
      TETIM=0.
      TF=0.
      TI=0.
      TETI3=0.
      DT=2.5
      ALLAT=39.5882
      ALLNG=-84.0829
      ALRAD=6369762.
      RAD=57.29578
      DO 1 I=1,12
      XC(I)=0.
      YC(I)=0.
      ZC(I)=0.
      XCA(I)=0.
      YCA(I)=0.
      ZCA(I)=0.
      XCB(I)=0.
      YCB(I)=0.
    1 ZCB(I)=0.
```

```
KK=0
IVAR=0
XSAT=0.
YSAT=0.
ZSAT=0.
XAL=0.
YAL=0.
ZAL=0.
XX=0.
YY=0.
ZZ=0.
XXA=0.
YYA=0.
ZZA=0.
AZI=0.
ELEV=0.
AZIIN=0.
ELEIN=0.
SDBIT=0.549325E-02
DABIT=0.03125
ERA=0.
ERE=0.
SLAT=0.
SLNG=0.
CLAT=0.
CLNG=0.
K=0
L=0
M=0
AZBIAS=0.
ELBIAS=0.
TIBIAS=0.
XSATC=0.
YSATC=0.
ZSATC=0.
DTC=0.
XDOTI=0.
YDOTI=0.
ZDOTI=0.
XDOT=0.
YDOT=0.
FREQ=0.
DOPLER=0.
TDLAY=0.
PP=0.
TC=0.
TK=0.
PW=0.
AN=0.
IOUNT=1
INTFLG=0
TMID=21600.
T=SECNDS(0.)
IF(T.GE.43200.)TMID=64800.
```

```
      CALL INITAL
      IF(IEOF.EQ.0)GO TO 2
      WRITE(6,52)
      GO TO 10
   52 FORMAT('    CAN NOT LOCATE EPHEMERIS DATA. RAPP CANCELLED.')
    2 CALL FIXMEM(RTLOOP,)
      CALL FIXMEM (DOIT,)

      CALL RUN(DOIT,,12,1,12,1,JJ)
      IF (JJ-1)21,11,21
   21 WRITE(6,51)JJ
   51 FORMAT('    DOIT START ERROR',2X,I6)
      GO TO 10
   11 CALL RUN(RTLOOP,,1,2,1,2,JJ)
      IF(JJ-1)20,10,20
   20 WRITE(6,50)JJ
   10 CALL UNFIX(RAPP,)
      CALL EXIT
  100 FORMAT(33H ROOFTOP ANTENNA POINTING PROGRAM)
  101 FORMAT(33H SET ALL CONSOLE SWITCHES TO ZERO)
   50 FORMAT('    RTLOOP START ERROR',2X,I6)
      END
```

SUBROUTINE INITAL

```
      SUBROUTINE INITAL
C
C        THIS ROUTINE PERFORMS ALL NECESSARY CALCULATIONS AND SETS
C        ALL VARIABLES TO THE REQUIRED STATUS
C
      COMMON /AL/ALLAT,ALLNG,ALRAD,RAD,SLAT,SLNG,CLAT,CLNG
      COMMON /COORD/XSAT,YSAT,ZSAT,XAL,YAL,ZAL,XX,YY,ZZ,XXA,YYA,ZZA
      COMMON /FDUM1/ALAT,F2,F3,F4,F5,F6,F7,F8,F9,F10
      DATA ASMINT,STATE1,PNT1/6RASMINT,6RSTATE1,6RPNT1  /
C        INITIALIZE DEVICES
      CALL START (ASMINT,0,0,JJ)
      IF(JJ-1)10,10,11
   11 WRITE(6,50)
      STOP
C        CONVERT LONGITUDE AND LATITUDE TO RADIANS
   10 CALL DATAIO
      ALLAT=ALAT/RAD
      ALAT=90.0-ALAT
      CLLAT=ALAT/RAD
      ALLNG=ALLNG/RAD
C        COMPUTE SINE AND COSINE OF LATITUDE AND LONGITUDE
      SLAT=SIN(ALLAT)
      CLAT=COS(ALLAT)
   50 FORMAT('  ASMINT START ERROR')
      SLNG=SIN(ALLNG)
      CLNG=COS(ALLNG)
C        COMPUTE LAB POSITION IN GEOCENTRIC COORDINATES
      XAL=CLAT*CLNG*ALRAD
      YAL=CLAT*SLNG*ALRAD
      ZAL=SLAT*ALRAD
      SLAT = SIN(CLLAT)
      CLAT = COS(CLLAT)
      RETURN
      END
```

SUBROUTINE DATAIO

```
      SUBROUTINE DATAIO
C        IDAY - DAY OF THE YEAR STARTING FROM 1 JAN. - INTEGER
C        DTC - MIT TIME BIAS IN SECONDS - REAL
C        TBID - MIDTIME OF THE CURRENT PERIOD IN HOURS - REAL
C        FREQ - CARRIER FREQUENCY IN HZ - REAL
C        II - 1 TO INITALIZE THE DISK FILE, 0 TO LEAVE AS IS - INTEGER
C        LES - NUMBER TO INDICATE SATELLITE UNDER USE (8 OR 9) - INTEGER
C     RH =RELATIVE HUMIDITY
C     PP =ABSOLUTE PRESSURE IN MILLIBARS
C     TFAR =TEMPERATURE IN DEGREES F
      DIMENSION IARAY(350)
      COMMON /C/XC(12),YC(12),ZC(12),XCA(12),YCA(12),ZCA(12)
      COMMON /C/XCB(12),YCB(12),ZCB(12),KK,IVAR
      COMMON /FL/K,L,M
      COMMON /SATC/XSATC,YSATC,ZSATC,DTC
      COMMON /BIAS/AZBIAS,ELBIAS,TIBIAS
      COMMON /TDDOP/FREQ,DOPLER,TDLAY
      COMMON /PTHN/PP,TC,TK,PW,AN
      COMMON /IDUM1/LES,I2,IEOF,I4,I5,IDAY,IERFLG,I8,I9,I10
      COMMON /AL/ALLAT,ALLNG,ALRAD,RAD,SLAT,SLNG,CLAT,CLNG
      COMMON /TT/TMID,TETIM,TF,TI,TETI3,DT
      COMMON /FDUM1/ALAT,F2,F3,F4,F5,F6,F7,F8,F9,F10
      K = 1
      WRITE(5,100)
      READ (1.101) IDAY,DTC,FREQ,II,LES
C        READ REFRACTION INPUTS AND PREPARE THEM
      READ(1,112)PP,TFAR,RH,AZBIAS,ELBIAS,ALLAT,ALLNG
      RLAT=ALLAT/RAD
      ALAT=ALLAT-.19243*SIN(2*RLAT)+.00032314*SIN(4.*RLAT)
     1-.0000007222*SIN(6.*RLAT)
      ALRAD=6378160.*(.998327073+.001676438*COS(2.*RLAT)
     1-.000003519*COS(4.*RLAT)+.000000008*COS(6.*RLAT))
      PP=PP*33.8636
      TC=(TFAR-32.)*5./9.
      TK=TC+273.18
      AK1=77.6
      AK2=0.
      AK3=4810.
      TWV=TFAR+460.
      IF(TFAR.LT.32.0)GO TO 30
      PW=RH*1.53197*EXP(16.17916*(TWV-590.)/TWV)
      GO TO 40
   30 PW=RH*.061003*EXP(22.51093*(TWV-492.)/TWV)
   40 AN=AK1*PP/TK+AK2*PW/TK+AK3*PW/TK**2
      WRITE(5,105) LES
      WRITE(5,102) IDAY
      WRITE(5,104) DTC
      KK = 0
      WRITE(6,103)LES
  103 FORMAT(1H ,///1X,'POINTING TO LES',I2///)
    3 CALL ASSIGN(3,'DK0:DAVEM.DAT;1',15)
      IFLG=0
      JFLG=0
      DEFINE FILE 3(1000,345,U,IVAR)
```

```
    2 KK = KK + 1
    7 READ(3'KK,END=14)(IARAY(I),I=1,345)
      DECODE (679,106,IARAY) LLES,(XCA(I),I=1,10),(YCA(I),I=1,10),
     1    (ZCA(I),I=1,10),JDAY
      IF(JDAY-IDAY)8,9,8
    8 KK=KK+1
      GO TO 7
    9 KK = KK + 1
      IF(TMID.EQ.21600.)GO TO 5
      IF(JFLG.NE.0)GO TO 5
      JFLG=1
      GO TO 7
    5 IF(LES.EQ.8)GO TO 6
      IF(IFLG.NE.0)GO TO 6
      KK=KK+3
      IFLG=1
      JFLG=0
      GO TO 7
    6 READ(3'KK,END=14)(IARAY(I),I=1,345)
      DECODE (679,106,IARAY) LLES,(XCB(I),I=1,10),(YCB(I),I=1,10),
     1    (ZCB(I),I=1,10),JDAY
      WRITE(5,107)
      IF(TMID.EQ.21600.)WRITE(5,108)JDAY
      J=JDAY-1
      IF(TMID.NE.21600.)WRITE(5,111)J
      DO 12 I=1,10
      WRITE(5,109) XCA(I),YCA(I),ZCA(I)
   12 CONTINUE
      WRITE(5,110)
      IF(TMID.EQ.21600.)WRITE(5,111)JDAY
      IF(TMID.NE.21600.)WRITE(5,108)JDAY
      DO 13 I=1,10
      WRITE(5,109) XCB(I),YCB(I),ZCB(I)
   13 CONTINUE
      IF(TMID.EQ.21600.)GO TO 10
      DO 11 I=1,10
      X=XCA(I)
      Y=YCA(I)
      Z=ZCA(I)
      XCA(I)=XCB(I)
      YCA(I)=YCB(I)
      ZCA(I)=ZCB(I)
      XCB(I)=X
      YCB(I)=Y
   11 ZCB(I)=Z
   10 WRITE(5,110)
      WRITE(5,113)PP
      WRITE(5,114)PW
      WRITE(5,115)TFAR
      WRITE(5,202)
  202 FORMAT(1H1)
      RETURN
C         SET END OF FILE FLAG
   14 IEOF=1
```

```
      RETURN
  100 FORMAT (33H ROOFTOP ANTENNA POINTING PROGRAM,/)
  101 FORMAT (I3,3XE14.7,3XE14.7,3XI1,3XI1)
  102 FORMAT (12H DAY NUMBER I3)
  104 FORMAT (34H THE MIT BIAS FOR THIS PERIOD IS  E15.8,9H  SECONDS,/)
  105 FORMAT (15H SATELLITE LES-I1)
  106 FORMAT (10XI1,7XE16.10,29(6XE16.10),4XI3)
  107 FORMAT (17H    X COEFFICIENTS,11X14HY COEFFICIENTS,
     1  11X14HZ COEFFICIENTS)
  108 FORMAT(30X,'A.M. EPHEMERIS FOR DAY ',I4)
  109 FORMAT(2XE17.10,8X,E17.10,8X,E17.10)
  110 FORMAT (1H )
  111 FORMAT(30X,'P.M. EPHEMERIS FOR DAY ',I4)
  112 FORMAT(7F10.4)
  113 FORMAT (17H ABS. PRESSURE = E15.8,10H MILLIBARS)
  115 FORMAT (15H TEMPERATURE = E15.8,7H DEGS F)
  114 FORMAT (32H WATER VAPOR PARTIAL PRESSURE = E15.8,10H MILLIBARS)
      END
```

TASK ASMINT

```
        .TITLE ASMINT
        .GLOBL ASMINT
        .MCALL EXIT$C
        .PSECT
        PC=%7
ASMINT: MOV #1,@#167760
        MOV #1,@#167750
1$: TST @#167770    ; TEST THAT REQ B IS RESET
BPL 2$              ; ITHEN BRANCH OUT
MOV @#167774,R0     ; ELSE READ DR-11 TO RESET REQ B
BR 1$
2$: MOV #3775,@#167762
        MOV #4000,@#167752
        EXIT$C
        .END ASMINT
```

TASK DOIT

```
        .TITLE DOIT
        .GLOBL DOIT,STATE,PNT,ANTENA
        .MCALL EXIT$C
        .PSECT ANTC,GBL,OVR
        COUNT=.+0
        ANTFLG=.+2
        .=.+4
        .PSECT MRTL
        PC=%7
DOIT:   JSR PC,STATE
        JSR PC,PNT
        DEC COUNT
        TST COUNT
        BNE OUT
        MOV #1,ANTFLG           ;SET ANTENNA FLAG
         MOV #31,COUNT          ;CALL ANTENNA EVERY 5 SECONDS
        JSR PC,ANTENA
  OUT:  EXIT$C MRTL
        .END DOIT
```

SUBROUTINE STATE

```
        .TITLE STATE
;       THIS ROUTINE READS THE S/D'S AND DEVELOPS THE PROPER
;       FLOATING POINT NUMBER
        .GLOBL STATE
        .PSECT ANT,GBL,OVR
        AZI=.+0
        ELEV=.+4
        AZIIN=.+10
        ELEIN=.+14
        SDBIT=.+20
        DABIT=.+24
        ERA=.+30
        ERE=.+34
        .=.+40
        .PSECT MRTL
        R0=%0
        R1=%1
        SP=%6
        PC=%7
STATE:  SETF
        SETI
        MOV R0,-(SP)
        MOV R1,-(SP)
        CLR R0
        CLR R1
        BIC #1,@#167760         ;INHIBIT AZI S/D
        BIC #1,@#167750         ;INHIBIT ELEV S/D
        LDF SDBIT,%2
        CLRF %0
        CLRF %1
        MOV @#167764,R0         ;READ AZIMUTH
        MOV @#167754,R1         ;READ ELEVATION
        ADD #1,@#167760         ;ENABLE AZI S/D
        ADD #1,@#167750         ;ENABLE ELEV S/D
        TST R0
        BGE OK
        LDCIF OVER,%1
        BIC #100000,R0
    OK: LDCIF R0,%0
        MULF %2,%0
        ADDF %1,%0
        STF %0,AZIIN
        CLRF %0
        LDCIF R1,%0
        MULF %2,%0
        STF %0,ELEIN
        MOV (SP)+,R1
        MOV (SP)+,R0
        RTS PC
 OVER:  .WORD 180.
        .END
```

SUBROUTINE PNT

```
        .TITLE PNT
        .GLOBL PNT
;
;               B HOLSAPPLE
;
        .PSECT ANT,GBL,OVR
        AZI=.+0
        ELEV=.+4
        AZIIN=.+10
        ELEIN=.+14
        SDBIT=.+20
        DABIT=.+24
        ERA=.+30
        ERE=.+34
        .=.+40
        .PSECT MRTL
   SW:  .WORD 0
 HOLD:  .WORD 0
 CON1:  .WORD 64.
 CON2:  .WORD -10.
 CON3:  .WORD 360.
 CON4:  .WORD 59.
 CON5:  .WORD 10.
  PNT:  SETF
        SETI
        CLRF %0
        CLRF %1
        CLRF %2
        CLRF %3
        LDF AZI,%0              ;ADD 59 DEG TO BOTH AZIMUTHS
        LDCIF CON4,%3
        ADDF %3,%0
        LDF AZIIN,%1
        ADDF %3,%1
        LDCIF CON3,%3
        CMPF %0,%3              ;REDUCE ANY AZIMUTHS EXCEEDING 360 DEG
        CFCC
        BLT E
        SUBF %3,%0
    E:  CMPF %1,%3
        CFCC
        BLT D
        SUBF %3,%1
    D:  SUBF %1,%0              ;SUBTRACT ACTUAL AZIMUTH FROM DESIRED AZIMUTH
        STF %0,ERA              ;STORE DIFFERENCE
        LDF ELEV,%0
        LDF ELEIN,%1
        SUBF %1,%0              ;SUBTRACT ACTUAL ELEV FROM DESIRED ELEV
        STF %0,ERE
        MOV #1,SW               ;SET RETURN
        LDF ERA,%2
        BR A                    ;GO ENCODE AZI CORRECTION FOR DR11C
    B:  MOV HOLD,@#167762       ;UPDATE D-11C FO- A0& CO--ECT&ON
        MOV #2,SW               ;SET RETURN
```

```
    LDF ERE,%2
    BR A                      ;GO ENCODE ELE CORRECTION FOR DR11C
C:  MOV HOLD,@#167752         ;UPDATE DR11C FOR ELE CORRECTION
    BR H
A:  LDCIF CON2,%3
    CMPF %2,%3                ;IF CORR IS LESS THAN -10 DEG MOVE 3300 OCTAL
    CFCC
    BGT F
    MOV #3300,HOLD
    BR I
F:  LDCIF CON5,%3
    CMPF %2,%3                ;IF CORRECTION IS MORE THAN +10 DEG MOVE 4500 OCTAL
    CFCC
    BLT G
    MOV #4500,HOLD
    BR I
G:  LDCIF CON1,%3
    ADDF %3,%2                ;ADD 64 TO CORRECTION
    LDF DABIT,%3              ;LOAD 0.03125 TO 3RD ACCUMULATOR
    DIVF %3,%2
    STCFI %2,HOLD             ;INTEGERIZE QUOTIENT AND STORE
I:  CMP #1,SW
    BEQ B                     ;RETURN LOGIC
    BR C
H:  RTS %7
    .END
```

SUBROUTINE ANTENA

```
      SUBROUTINE ANTENA
C         THIS SUBROUTINE COMPUTES SATELLITE POSITION AND DESIRED
C         ANTENNA AZIMUTH AND ELEVATION
      COMMON /TT/TMID,TETIM,TF,TI,TETI3,DT
      COMMON /AL/ALLAT,ALLNG,ALRAD,RAD,SLAT,SLNG,CLAT,CLNG
      COMMON /C/XC(12),YC(12),ZC(12),XCA(12),YCA(12),ZCA(12)
      COMMON /C/XCB(12),YCB(12),ZCB(12),KK,IVAR
      COMMON /COORD/XSAT,YSAT,ZSAT,XAL,YAL,ZAL,XX,YY,ZZ,XXA,YYA,ZZA
      COMMON /ANT/AZI,ELEV,AZIIN,ELEIN,SDBIT,DABIT,ERA,ERE
      COMMON /BIAS/AZBIAS,ELBIAS,TIBIAS
      COMMON /SATC/XSATC,YSATC,ZSATC,DTC
C         READ TIME FROM THE CLOCK
      CALL CTIME
C         COMPUTE SATELLITE POSITION
      XSAT=0.0
      YSAT=0.0
      ZSAT=0.0
      DO 1 I=1,9
      J=11-I
      XSAT=(XSAT+XC(J))*TETI3
      YSAT=(YSAT+YC(J))*TETI3
    1 ZSAT=(ZSAT+ZC(J))*TETI3
      XSAT=XSAT+XC(1)+XSATC
      YSAT=YSAT+YC(1)+YSATC
      ZSAT=ZSAT+ZC(1)+ZSATC
C         COMPUTE VECTOR FROM AVIONICS LAB TO SATELLITE
      XX=XSAT-XAL
      YY=YSAT-YAL
      ZZ=ZSAT-ZAL
C         ROTATE VECTOR INTO AVIONICS LAB COORDINATES
      XXA=CLAT*CLNG*XX+CLAT*SLNG*YY-SLAT*ZZ
      YYA=-SLNG*XX+CLNG*YY
      ZZA=SLAT*CLNG*XX+SLAT*SLNG*YY+CLAT*ZZ
C         COMPUTE AZIMUTH AND ELEVATION FOR ANTENNA
      AXXA=ABS(XXA)
      AYYA=ABS(YYA)
      AZZA=ABS(ZZA)
      AXYA=SQRT(XXA*XXA+YYA*YYA)
      AZI=ATAN2(AYYA,AXXA)
      ELEV=ATAN2(AZZA,AXYA)
C         PUT AZIMUTH INTO CORRECT QUADRANT
      IF(XXA)10,12,12
   10 IF (YYA)13,14,15
   13 AZI=(360./RAD)-AZI
      GO TO 25
   14 AZI=0.0
      GO TO 25
   15 AZI=AZI
      GO TO 25
   12 IF(YYA)16,17,18
   16 AZI=(180./RAD)+AZI
      GO TO 25
   17 AZI=180./RAD
      GO TO 25
```

```
   18 AZI=(180./RAD)-AZI
C        TEST TO KEEP ANTENNA FROM POINTING AT THE TOWERS (276 TO 302 DEG)
   25 TEST1=276./RAD
      TEST2=302./RAD
      TEST3=288./RAD
      IF(AZI.GT.TEST1) GO TO 26
      GO TO 30
   26 IF (AZI.LT.TEST2) GO TO 27
      GO TO 30
   27 IF(AZI.LE.TEST3) GO TO 28
      AZI=TEST2
      GO TO 30
   28 AZI=TEST1
   30 AZI=AZI*RAD+AZBIAS
      ELEV=ELEV*RAD+ELBIAS
      IF(ZZA.LT.0.0)ELEV=-ELEV
      CALL REFCT
      RETURN
      END
```

SUBROUTINE CTIME

```
      SUBROUTINE CTIME
C        THIS ROUTINE PROVIDES TIME FOR THE SATELLITE POSITION CALCULATION
      COMMON /TT/TMID,TETIM,TF,TI,TETI3,DT
      COMMON /BIAS/AZBIAS,ELBIAS,TIBIAS
      COMMON /C/XC(12),YC(12),ZC(12),XCA(12),YCA(12),ZCA(12)
      COMMON /C/XCB(12),YCB(12),ZCB(12),KK,IVAR
      CALL TOD
      TET=SECNDS(0.)
      CHKT=TETIM-TET
      CHKT=ABS(CHKT)
      IF(CHKT-10.)1,2,2
    2 TETIM=TET
    1 IF(TETIM.LT.43200.)GO TO 4
      DO 10 I=1,10
      XC(I)=XCB(I)
      YC(I)=YCB(I)
   10 ZC(I)=ZCB(I)
      TMID=64800.
      TI=43200.
      TF=86400.
      TETI3=(TETIM+DT+TIBIAS-TMID)/21600.
      RETURN
    4 DO 20 I=1,10
      XC(I)=XCA(I)
      YC(I)=YCA(I)
   20 ZC(I)=ZCA(I)
      TMID=21600.
      TI=0.0
      TF=43200.
      TETI3=(TETIM+DT+TIBIAS-TMID)/21600.
      RETURN
      END
```

SUBROUTINE TOD

```
.TITLE    TOD
.GLOBL TOD
.PSECT    TT,GBL,OVR
TMID=.+0
TETIM=.+4
TF=.+10
TI=.+14
TETI3=.+20
DT=.+24
.=.+30
.PSECT    MRTL
R0=%0
AC0=%0
R1=%1
AC1=%1
R2=%2
AC2=%2
BUF=%2
R3=%3
NEG1=%4
NEG2=%4
R5=%5
WORD:     .BLKW     2
CON1:     .WORD     1000.
CON2:     .WORD     10.
ADD0:     167770                        ; CONTROL & STATUS
ADD4:     167774                        ; DATA IN
TOD:      MOV       R0,-(SP)
          MOV       R1,-(SP)
          MOV       R2,-(SP)
          MOV       BUF,-(SP)
         MOV       NEG1,-(SP)
          MOV       R5,-(SP)
          SETI
          SETF
          MOV       #1,@ADD0            ; SET CSR0
NOP
NOP
          NOP
          CLR       @ADD0               ; RESET CSR0
INP1:     TST       @ADD0               ; TEST REQ B
          BPL       INP1
MOV #10.,R0
I1: NOP
SOB R0,I1
          MOV       @ADD4,WORD          ; INPUT BYTE1
INP2:     TST       @ADD0               ; TEST THAT REQ B IS RESET
          BMI       INP2
          MOV       #1,@ADD0            ; SET CSR0
          NOP
NOP
NOP
          CLR       @ADD0               ; RESET CSR0
INP3:     TST       @ADD0               ; TEST REQ B
```

```
        BPL     INP3
MOV #10.,R0
I2: NOP
SOB R0,I2
        MOV     @ADD4,WORD+2            ; INPUT BYTE2
; DECODE TIME WORDS
; R1= QUOTIENT
; R2= REMAINDER
        CLR     NEG1
        TST     WORD                    ; CHECK IF MINUS
        BPL     W1
        MOV     #8000.,NEG1             ; 8000 MSEC = 8 SEC
        BIC     #100000,WORD
W1:     CLR     R0                      ; SET UP DIVIDEND
        MOV     WORD,R1                 ; SET UP DIVIDEND
        DIV     #4096.,R0
        MOV     R0,R3
        MUL     #1000.,R3
        MOV     R3,BUF
        CLR     R0
        DIV     #256.,R0                ; DIVIDE REMAINDER
        MOV     R0,R3
        MUL     #100.,R3
        ADD     R3,BUF
        CLR     R0
        DIV     #16.,R0                 ; DIVIDE REMAINDER
        MOV     R0,R3
        MUL     #10.,R3
        ADD     R3,BUF
        ADD     R1,BUF                  ; ADD REMAINDER
        ADD     NEG1,BUF                ; ADD IN BIT 15 (NORMALLY SIGN BIT)
        ADD     #36,BUF
; BUF NOW CONTAINS MSEC
        LDCIF   CON1,AC1
        LDCIF   BUF,AC0
        DIVF    AC1,AC0
; AC0 NOW CONTAINS SECONDS
        CLR     NEG2
        TST     WORD+2                  ; CHECK IF MINUS
        BPL     W2
        MOV     #7200.,NEG2             ; 7200(10 SEC) = 20 HOURS
        BIC     #100000,WORD+2
W2:     CLR     R0                      ; SET UP DIVIDEND
        MOV     WORD+2,R1               ; SET UP DIVIDEND
        DIV     #16384.,R0
        MOV     R0,R3
        MUL     #3600.,R3
        MOV     R3,BUF
        CLR     R0
        DIV     #1024.,R0               ; DIVIDE REMAINDER
        MOV     R0,R3
        MUL     #360.,R3
        ADD     R3,BUF
        CLR     R0
```

```
        DIV     #128.,R0            ; DIVIDE REMAINDER
        MOV     R0,R3
        MUL     #60.,R3
        ADD     R3,BUF
        CLR     R0
        DIV     #8.,R0              ; DIVIDE REMAINDER
        MOV     R0,R3
        MUL     #6,R3
        ADD     R1,BUF              ; ADD REMAINDER
        ADD     R3,BUF
        ADD     NEG2,BUF
; BUF NOW CONTAINS 10 SECONDS
        LDCIF   CON2,AC2
        LDCIF   BUF,AC1
        MULF    AC2,AC1
        ADDF    AC1,AC0
; AC0 NOW CONTAINS SECONDS IN FLOATING POINT
        STF     AC0,TETIM
        MOV     (SP)+,R5
        MOV     (SP)+,NEG1
        MOV     (SP)+,BUF
        MOV     (SP)+,R2
        MOV     (SP)+,R1
        MOV     (SP)+,R0
        RTS     %7
        .END
```

SUBROUTINE REFCT

```
      SUBROUTINE REFCT
C        PRESSURE,TEMPERATURE,HUMIDITY AND REFRACTIVITY
      COMMON /PTHN/PP,TC,TK.PW,AN
      COMMON /ANT/AZI,ELEV.AZIIN,ELEIN,SDBIT,DABIT,ERA,ERE
C     PP =ABSOLUTE PRESSURE IN MILLIBARS
C     TC =TEMPERATURE IN DEGREES CENTIGRADE
C     TK =TEMPERATURE IN DEGREES KELVIN    =TC+273.18
C     PW =WATER VAPOR PARTIAL PRESSURE IN MILLIBARS
C     AN =SURFACE REFRACTIVITY = (REFRACTIVE INDEX-1.)*1.E6
C     A,B=REFRACTION TABLE
C
C        BYPASS CORRECTION IF INDEX SET TO ZERO.
C     IF(AN .EQ. 0.00) D1=D3=0.00; RETURN
      IF(AN .NE. 0.00) GO TO 50
      D1    =0.00
      D3    =0.00
      RETURN
   50 CONTINUE
C
C        ENTER WITH ELEVATION IN DEGREES --- IF EL IS APPARENT, THEN
C        TRUE = (APPARENT - D1).    IF EL IS TRUE, THEN
C        APPARENT = (TRUE + D3).    AN IS SURFACE REFRACTIVITY (N-UNITS).
C
C        NOMINAL VALUES OF TEMP (T), PRESSURE (P), AND WATER VAPOR
C        PRESSURE (PWV) ARE T=283, P=758, AND PWV=6 .
C     AN = REFRACTIVITY      = K1*P/T+K2*PWV/T+K3*PWV/T**2
C        K1=77.6,    K2=0.,    K3=4810.
C
C     *    *    *    *    *    A FREED  MILLSTONE RADAR    *    *    *
      ELTRUE=ELEV
      IF(ELTRUE.LT.0.) ELTRUE=0.0
      CALL DELL(ELTRUE,D1)
      E1=ELTRUE+D1
      CALL DELL(E1,D2)
      E2=ELTRUE+D2
      CALL DELL(E2,D3)
      ELEV=ELTRUE+D3
      RETURN
      END
```

SUBROUTINE DELL

```
C       A.FREED   DEC 1972   SUBROUTINE DELL (MILLSTONE REFRACTION TABLE)
        SUBROUTINE DELL(E,D)
C          PRESSURE,TEMPERATURE,HUMIDITY AND REFRACTIVITY
        COMMON /PTHN/PP,TC,TK,PW,AN
        DIMENSION A(29),B(29)
C       PP =ABSOLUTE PRESSURE IN MILLIBARS
C       TC =TEMPERATURE IN DEGREES CENTIGRADE
C       TK =TEMPERATURE IN DEGREES KELVIN    =TC+273.18
C       PW =WATER VAPOR PARTIAL PRESSURE IN MILLIBARS
C       AN =SURFACE REFRACTIVITY = (REFRACTIVE INDEX-1.)*1.E6
C       A,B=REFRACTION TABLE
C
C           TABLE IS SET UP IN FORM A+B*NS FOR A SEQUENCE OF FIXED ELEVATIO
C           ANGLES WHERE NS = SURFACE REFRACTIVITY (N-UNITS) = AN BASED
C           ON ALBANY WEATHER BUREAU DATA FOR FEBRUARY AND AUGUST.
C
C
C           ENTRIES A (MDEG) AND B (MDEG/N-UNIT) CORRESPOND TO ELEVATION
C           INTERVALS OF Ø.5 DEG BETWEEN Ø DEG AND 6 DEG ELEVATION,
C           INTERVALS OF 2.Ø DEG BETWEEN 6 DEG AND 2Ø DEG ELEVATION,
C           INTERVALS OF 1Ø. DEG BETWEEN 2Ø DEG AND 9Ø DEG ELEVATION.
C           MODEL FAILS IF UNREALISTIC REFRACTIVITY IS SPECIFIED.
C
        DATA A/427.5,388.7,268.3,16Ø.,95.9,6Ø.,4Ø.98,3Ø.,2Ø.,14.,1Ø.23,7.,
       +3.,1.3,.3Ø5,-.1,-.3.-.5,-.54,-.56,-.5,-.35,-.24,-.18,-.12,-.Ø5,Ø.,
       +Ø.,Ø./
        DATA B/3.825,3.127.2.371,1.78,1.4Ø9,1.16,.985,.87,.77,.68,.6Ø9,
       +.555,.511,.386,.3Ø9,.256,.218,.19Ø,.168,.15,.Ø95,.Ø653,.Ø461,
       +.Ø316,.Ø199,.ØØ97,Ø.,Ø.,Ø./
        IF (E .GT. 2Ø.) GO TO 2Ø
        IF (E .GT. 6.) GO TO 6
        IX=1+INT(2.*E)
        DEL=(AMOD(E,Ø.5))*2.Ø
      2 AA=A(IX)-(A(IX)-A(IX+1))*DEL
        BB=B(IX)-(B(IX)-B(IX+1))*DEL
        D=.ØØ1*(-AA+BB*AN)
        GO TO 99
     2Ø IX=18+INT(E/ 1Ø.)
        DEL=(AMOD(E,1Ø.))*.1
        GO TO 2
      6 IX=13+INT((E-6.)/2.)
        DEL=(AMOD(E,2.))*.5
        GO TO 2
     99 RETURN
        END
```

```
        .TITLE RTLOOP
        .GLOBL RTLOOP,AEOUT
        .MCALL EXIT$C, RUN$C
        .PSECT ANTC,GBL,OVR
        COUNT=.+0
        ANTFLG=.+2
        .=.+4
.PSECT IDUM1,GBL,OVR
LES=.+0
I2=.+2
I3=.+4
I4=.+6
I5=.+8.
I6=.+10.
I7=.+12.
I8=.+14.
I9=.+16.
I10=.+18.
.=.+20.
        .PSECT MRTL
        R0=%0
        R1=%1
        R2=%2
        R3=%3
        PC=%7
RTLOOP: NOP
TST ANTFLG
BEQ OUT
MOV LES,I10
CLR ANTFLG
JSR PC,AEOUT
CMP LES,I10
BEQ OUT
RUN$C DATAIL,GEN,365,142,4,0,1,0,0,MRTL
OUT: EXIT$C MRTL
        .END
```

SUBROUTINE AEOUT

```
      SUBROUTINE AEOUT
      COMMON /COORD/XSAT,YSAT,ZSAT,XAL,YAL,ZAL,XX,YY,ZZ,XXA,YYA,ZZA
      COMMON /ANT/AZI,ELEV,AZIIN,ELEIN,SDBIT,DABIT,ERA,ERE
      COMMON /BIAS/AZBIAS,ELBIAS,TIBIAS
      COMMON /IDUM1/LES,N,IEOF,IAMFLG,IPMFLG,I6,IERFLG,I8,I9,I10
      Z=SQRT(XXA*XXA+YYA*YYA+ZZA*ZZA)
      Z=Z/1000.
      I=INT(Z-10000.)
      J=INT(Z/.3-131000.)
      READ(6,100)J1,J2,J3,J4,J5,J6
  100 FORMAT(6A1)
      IF(J1.NE.'I')GO TO 1
      WRITE(6,200)AZI,ELEV,Z,I,J,ERA,ERE
  200 FORMAT(1H ,'AZIMUTH= ',   F8.3/1X,   'ELEVATION= ',   F8.3/1X,
     1'RANGE= '    F8.1,1X,'(',O6,   ')'/1X,   'PATH DELAY= ',  '(',O6,   ')
     1'/1X,'AZIMUTH ERROR= ',  F6.3/1X,  'ELEVATION ERROR= ',   F8.3)
      GO TO 3
    1 IF(J1.NE.'A')GO TO 2
      CALL CONVER(J2,J3,J4,J5,J6,A)
      AZBIAS=A
      GO TO 3
    2 IF(J1.NE.'E')GO TO 4
      CALL CONVER(J2,J3,J4,J5,J6,A)
      ELBIAS=A
      GO TO 3
    4 IF(J1.NE.'L')GO TO 3
      LES=0
      IF(J2.EQ.'8')LES=8
      IF(J2.EQ.'9')LES=9
    3 RETURN
      END
```

SUBROUTINE CONVER

```
      SUBROUTINE CONVER(J2,J3,J4,J5,J6,A)
      KOUNT=1
      ISIGN=0
      IDOT=0
      NUM=0
      JJ=1
    1 IF(KOUNT.EQ.1)I=J2
      IF(KOUNT.EQ.2)I=J3
      IF(KOUNT.EQ.3)I=J4
      IF(KOUNT.EQ.4)I=J5
      IF(KOUNT.EQ.5)I=J6
      IF(I.EQ.'-')GO TO 2
      IF(I.EQ.'.')GO TO 3
      IF(I.EQ.'0')II=0
      IF(I.EQ.'1')II=1
      IF(I.EQ.'2')II=2
      IF(I.EQ.'3')II=3
      IF(I.EQ.'4')II=4
      IF(I.EQ.'5')II=5
      IF(I.EQ.'6')II=6
      IF(I.EQ.'7')II=7
      IF(I.EQ.'8')II=8
      IF(I.EQ.'9')II=9
      IF(I.EQ.' ')GO TO 5
      GO TO 4
    2 ISIGN=1
      GO TO 5
    3 IDOT=1
      GO TO 5
    4 IF(IDOT.EQ.1)JJ=JJ*10
      NUM=10*NUM+II
    5 KOUNT=KOUNT+1
      IF(KOUNT.LE.5)GO TO 1
      IF(ISIGN.EQ.1)NUM=-NUM
      A=FLOAT(NUM)/FLOAT(JJ)
      RETURN
      END
```

```
      DIMENSION IARAY(350)
      COMMON /C/XC(12),YC(12),ZC(12),XCA(12),YCA(12),ZCA(12)
      COMMON /C/XCB(12),YCB(12),ZCB(12),KK,IVAR
      COMMON /FL/K,L,M
      COMMON /SATC/XSATC,YSATC,ZSATC,DTC
      COMMON /BIAS/AZBIAS,ELBIAS,TIBIAS
      COMMON /TDDOP/FREQ,DOPLER,TDLAY
      COMMON /PTHN/PP,TC,TK,PW,AN
      COMMON /IDUM1/LES,I2,IEOF,I4,I5,IDAY,IERFLG,I8,I9,I10
      COMMON /AL/ALLAT,ALLNG,ALRAD,RAD,SLAT,SLNG,CLAT,CLNG
      COMMON /TT/TMID,TETIM,TF,TI,TETI3,DT
      COMMON /FDUM1/ALAT,F2,F3,F4,F5,F6,F7,F8,F9,F10
      DATA DOIT,RTLOOP/6RDOIT  ,6RRTLOOP/
      CALL CANALL (DOIT,)
      CALL CANALL (RTLOOP,)
      K = 1
      KK = 0
    3 CALL ASSIGN(3,'DK0:DAVEM.DAT;1',15)
      IFLG=0
      JFLG=0
      DEFINE FILE 3(1000,345,U,IVAR)
    2 KK = KK + 1
    7 READ(3'KK,END=14)(IARAY(I),I=1,345)
      DECODE (679,106,IARAY) LLES,(XCA(I),I=1,10),(YCA(I),I=1,10),
     1   (ZCA(I),I=1,10),JDAY
      IF(JDAY-IDAY)8,9,8
    8 KK=KK+1
      GO TO 7
    9 KK = KK + 1
      IF(TMID.EQ.21600.)GO TO 5
      IF(JFLG.NE.0)GO TO 5
      JFLG=1
      GO TO 7
    5 IF(LES.EQ.8)GO TO 6
      IF(IFLG.NE.0)GO TO 6
      KK=KK+3
      IFLG=1
      JFLG=0
      GO TO 7
    6 READ(3'KK,END=14)(IARAY(I),I=1,345)
      DECODE (679,106,IARAY) LLES,(XCB(I),I=1,10),(YCB(I),I=1,10),
     1   (ZCB(I),I=1,10),JDAY
      IF(TMID.EQ.21600.)GO TO 10
      DO 11 I=1,10
      X=XCA(I)
      Y=YCA(I)
      Z=ZCA(I)
      XCA(I)=XCB(I)
      YCA(I)=YCB(I)
      ZCA(I)=ZCB(I)
      XCB(I)=X
      YCB(I)=Y
   11 ZCB(I)=Z
   10 CALL RUN(DOIT,,12,1,12,1,JJ)
```

```
      IF(JJ-1)1,4,1
    1 WRITE(6,200)JJ
  200 FORMAT('    DOIT START ERROR    ',I6)
      GO TO 14
    4 CALL RUN(RTLOOP,,1,2,1,2,JJ)
      IF(JJ-1)12,14,12
   12 WRITE(6,201)JJ
  201 FORMAT('    RTLOOP START ERROR    ',I6)
C         SET END OF FILE FLAG
   14 IEOF=1
      CALL EXIT
  106 FORMAT (10XI1,7XE16.10,29(6XE16.10),4XI3)
      END
```

TASK SCAN

```
      COMMON /TT/TMID,TETIM,TF,TI,TETI3,DT
      COMMON /AL/ALLAT,ALLNG,ALRAD,RAD,SLAT,SLNG,CLAT,CLNG
      COMMON /C/XC(12),YC(12),ZC(12),XCA(12),YCA(12),ZCA(12)
      COMMON /C/XCB(12),YCB(12),ZCB(12),KK,IVAR
      COMMON /COORD/XSAT,YSAT,ZSAT,XAL,YAL,ZAL,XX,YY,ZZ,XXA,YYA,ZZA
      COMMON /ANT/AZI,ELEV,AZIIN,ELEIN,SDBIT,DABIT,ERA,ERE
      COMMON /FL/K,L,M
      COMMON /BIAS/AZBIAS,ELBIAS,TIBIAS
      COMMON /SATC/XSATC,YSATC,ZSATC,DTC
      COMMON /DOTI/XDOTI,YDOTI,ZDOTI,XDOT,YDOT
      COMMON /TDDOP/FREQ,DOPLER,TDLAY
      COMMON /PTHN/PP,TC,TK,PW,AN
      COMMON /ANTC/IOUNT,INTFLG
      COMMON /BBIAS/ABIAS,EBIAS
      DATA SCAN/6RSCAN  /
      DATA SCANR/6RSCANR /
      DATA DOIT/6RDOIT  /
      DATA RTLOOP /6RRTLOOP/
      INTFLG=0
      ABIAS=-2.5
      EBIAS=2.5
      CALL CANALL(DOIT,)
      CALL CANALL(RTLOOP,)
      CALL FIXMEM(SCANR,JJ)
      CALL RUN (SCANR,,12,1,12,1,JJ)
      WRITE(6,51)JJ
   51 FORMAT(1H ,'SCANR RUN STATUS=',I6)
      DO 10 I=1,25
    1 IF(INTFLG.EQ.1)GO TO 2
      CALL WAIT(5,1,JJ)
      GO TO 1
    2 INTFLG=0
   10 CONTINUE
      IF((ABS(ERA).GT..1) .OR.(ABS(ERE).GT..1))WRITE(6,200)ERA,ERE
      WRITE(6,203)
      DO 20 I=1,26
      A=(-1.)**(I-1)
      DO 30 IJ=1,5
      DO 40 IK=1,5
      ABIAS=ABIAS+.2*A
    3 IF(INTFLG.EQ.1)GO TO 4
      CALL WAIT(5,1,JJ)
      GO TO 3
    4 INTFLG=0
   40 CONTINUE
      AZ=AZI
      EL=ELEV
      WRITE(6,204)AZ,EL
   30 CONTINUE
      WRITE(6,202)
C ONE AZIMUTH SCAN COMPLETE
      EBIAS=EBIAS-.2
   20 CONTINUE
      CALL CANALL(SCANR, )
```

```
      CALL UNFIX(SCAN,)
      CALL UNFIX(SCANR, )
      CALL RUN(DOIT,,12,1,12,1,)
      CALL RUN (RTLOOP,,1,2,1,2,)
      WRITE(6,201)
  200 FORMAT(1H ,'UNABLE TO INITIALIZE SCAN, AZI ERROR=',F8.3,/1X,'ELEV
     1ERROR=',F8.3)
  201 FORMAT(1H 'SCAN COMPLETE.')
  202 FORMAT(1H ,'END OF ONE AZIMUTH SCAN')
  203 FORMAT(1H ,'INITIALIZATION PHASE COMPLETE.')
  204 FORMAT(2F8.3)
      ABIAS=0.
      EBIAS=0.
      CALL EXIT
      END
```

TASK SCANR

```
.TITLE   SCANR
.GLOBL   SCANR,STATE,PNT,ANTENA
.MCALL   EXIT$C
.PSECT   ANTC,GBL,OVR
         COUNT=.+0
         ANTFLG=.+2
         .=.+4
.PSECT   MRTL
PC=%7
SCANR:   JSR        PC,STATE
         JSR        PC,PNT
         JSR        PC,ANTENA
OUT:     MOV        #1,ANTFLG
         EXIT$C     MRTL
.END     SCANR
```

SUBROUTINE PANTEN

```
      SUBROUTINE ANTENA
C        THIS SUBROUTINE COMPUTES SATELLITE POSITION AND DESIRED
C        ANTENNA AZIMUTH AND ELEVATION
      COMMON /TT/TMID,TETIM,TF,TI,TETI3,DT
      COMMON /AL/ALLAT,ALLNG,ALRAD,RAD,SLAT,SLNG,CLAT,CLNG
      COMMON /C/XC(12),YC(12),ZC(12),XCA(12),YCA(12),ZCA(12)
      COMMON /C/XCB(12),YCB(12),ZCB(12),KK,IVAR
      COMMON /COORD/XSAT,YSAT,ZSAT,XAL,YAL,ZAL,XX,YY,ZZ,XXA,YYA,ZZA
      COMMON /ANT/AZI,ELEV,AZIIN,ELEIN,SDBIT,DABIT,ERA,ERE
      COMMON /BIAS/AZBIAS,ELBIAS,TIBIAS
      COMMON /SATC/XSATC,YSATC,ZSATC,DTC
      COMMON /BBIAS/ABIAS,EBIAS
C        READ TIME FROM THE CLOCK
      CALL CTIME
C        COMPUTE SATELLITE POSITION
      XSAT=0.0
      YSAT=0.0
      ZSAT=0.0
      DO 1 I=1,9
      J=11-I
      XSAT=(XSAT+XC(J))*TETI3
      YSAT=(YSAT+YC(J))*TETI3
    1 ZSAT=(ZSAT+ZC(J))*TETI3
      XSAT=XSAT+XC(1)+XSATC
      YSAT=YSAT+YC(1)+YSATC
      ZSAT=ZSAT+ZC(1)+ZSATC
C        COMPUTE VECTOR FROM AVIONICS LAB TO SATELLITE
      XX=XSAT-XAL
      YY=YSAT-YAL
      ZZ=ZSAT-ZAL
C        ROTATE VECTOR INTO AVIONICS LAB COORDINATES
      XXA=CLAT*CLNG*XX+CLAT*SLNG*YY-SLAT*ZZ
      YYA=-SLNG*XX+CLNG*YY
      ZZA=SLAT*CLNG*XX+SLAT*SLNG*YY+CLAT*ZZ
C        COMPUTE AZIMUTH AND ELEVATION FOR ANTENNA
      AXXA=ABS(XXA)
      AYYA=ABS(YYA)
      AZZA=ABS(ZZA)
      AXYA=SQRT(XXA*XXA+YYA*YYA)
      AZI=ATAN2(AYYA,AXXA)
      ELEV=ATAN2(AZZA,AXYA)
C        PUT AZIMUTH INTO CORRECT QUADRANT
      IF(XXA)10,12,12
   10 IF (YYA)13,14,15
   13 AZI=(360./RAD)-AZI
      GO TO 25
   14 AZI=0.0
      GO TO 25
   15 AZI=AZI
      GO TO 25
   12 IF(YYA)16,17,18
   16 AZI=(180./RAD)+AZI
      GO TO 25
   17 AZI=180./RAD
```

```
      GO TO 25
   18 AZI=(180./RAD)-AZI
C         TEST TO KEEP ANTENNA FROM POINTING AT THE TOWERS (276 TO 302 DEG)
   25 TEST1=276./RAD
      TEST2=302./RAD
      TEST3=288./RAD
      IF(AZI.GT.TEST1) GO TO 26
      GO TO 30
   26 IF (AZI.LT.TEST2) GO TO 27
      GO TO 30
   27 IF(AZI.LE.TEST3) GO TO 28
      AZI=TEST2
      GO TO 30
   28 AZI=TEST1
   30 IF(AZBIAS.GE.128.)AZBIAS=128.-AZBIAS
      IF(ELBIAS.GE.128.)ELBIAS=128.-ELBIAS
      AZB=.05*AZBIAS
      ELB=.05*ELBIAS
      AZB=AZB*COS(ELEV)+ABIAS
      AZI=AZI*RAD+AZB
      ELEV=ELEV*RAD+ELB+EBIAS
      CALL REFCT
      RETURN
      END
```

TASK HAWAII

```
C          HAWAII ANTENNA POINTING PROGRAM
C
C          THIS TASK IS INSTALLED UNDER THE RSX-11D OPERATING SYSTEM.
C          HAWAII GENERATES LOOK ANGLES TO LES 8 OR LES 9 FOR EVERY
C          HOUR, FROM ANY POINT ON THE EARTH'S SURFACE.
C
C          INPUT DATA HAS UNITS OF DEGREES, METERS, AND SECONDS
C
      COMMON /TT/TMID,TETIM,TF,TI,TETI3,DT
      COMMON /AL/ALLAT,ALLNG,ALRAD,RAD,SLAT,SLNG,CLAT,CLNG
      COMMON /C/XC(12),YC(12),ZC(12),XCA(12),YCA(12),ZCA(12)
      COMMON /C/XCB(12),YCB(12),ZCB(12),KK,IVAR
      COMMON /COORD/XSAT,YSAT,ZSAT,XAL,YAL,ZAL,XX,YY,ZZ,XXA,YYA,ZZA
      COMMON /ANT/AZI,ELEV,AZIIN,ELEIN,SDBIT,DABIT,ERA,ERE
      COMMON /FL/K,L,M
      COMMON /BIAS/AZBIAS,ELBIAS,TIBIAS
      COMMON /SATC/XSATC,YSATC,ZSATC,DTC
      COMMON /DOTI/XDOTI,YDOTI,ZDOTI,XDOT,YDOT
      COMMON /TDDOP/FREQ,DOPLER,TDLAY
      COMMON /PTHN/PP,TC,TK,PW,AN
      COMMON /ANTC/IOUNT,INTFLG
      COMMON /IDUM1/LES,N,IEOF,IAMFLG,IPMFLG,IDAY,IERFLG,I8,I9,I10
      DATA RTLOOP/6RRTLOOP/
      DATA RAPP,STATE1,PNT1,DOIT/6RRAPP  ,6RSTATE1,6RPNT1  ,6RDOIT  /
C          PROGRAM NAME AND INSTRUCTIONS

      WRITE(6,100)
      WRITE(6,101)
      CALL ASSIGN(3,'DK0:DAVEM.DAT;1',15)
      DEFINE FILE 3(1000,345,U,IVAR)
C          INITIALIZATION ROUTINE
      IDAY=0
      IEOF=0
      IAMFLG=0
      IPMFLG=0
      IERFLG=0
      TMID=0.
      TETIM=0.
      TF=0.
      TI=0.
      TETI3=0.
      DT=2.5
      ALLAT=39.5882
      ALLNG=-84.0829
      ALRAD=6369762.
      RAD=57.29578
      DO 1 I=1,12
      XC(I)=0.
      YC(I)=0.
      ZC(I)=0.
      XCA(I)=0.
      YCA(I)=0.
      ZCA(I)=0.
      XCB(I)=0.
```

```
      YCB(I)=0.
    1 ZCB(I)=0.
      KK=0
      IVAR=0
      XSAT=0.
      YSAT=0.
      ZSAT=0.
      XAL=0.
      YAL=0.
      ZAL=0.
      XX=0.
      YY=0.
      ZZ=0.
      XXA=0.
      YYA=0.
      ZZA=0.
      AZI=0.
      ELEV=0.
      AZIIN=0.
      ELEIN=0.
      SDBIT=0.549325E-02
      DABIT=0.03125
      ERA=0.
      ERE=0.
      SLAT=0.
      SLNG=0.
      CLAT=0.
      CLNG=0.
      K=0
      L=0
      M=0
      AZBIAS=0.
      ELBIAS=0.
      TIBIAS=0.
      XSATC=0.
      YSATC=0.
      ZSATC=0.
      DTC=0.
      XDOTI=0.
      YDOTI=0.
      ZDOTI=0.
      XDOT=0.
      YDOT=0.
      FREQ=0.
      DOPLER=0.
      TDLAY=0.
      PP=0.
      TC=0.
      TK=0.
      PW=0.
      AN=0.
      ICOUNT=1
      INTFLG=0
    2 TMID=21600.
```

```
      CALL INITAL
      IF(IEOF.EQ.0)GO TO 3
      WRITE(6,52)
      GO TO 69
   52 FORMAT('    CAN NOT LOCATE EPHEMERIS DATA, RAPP CANCELLED.')
    3 DO 30 I=1,24
      TETIM=3600.*I
      IF(I.EQ.24)TETIM=86340.
      JT=100*I
      IF(I.EQ.24)JT=2359
      CALL ANTENA
      ALAT=ALLAT*RAD
      ALNG=ALLNG*RAD
   30 WRITE(5,200)IDAY,LES,JT,ALAT,ALNG,AZI,ELEV
  200 FORMAT(1H ,I4,2X,I2,2X,I5,2X,F4.0,2X,F5.0,2X,F7.2,2X,F7.2)
      WRITE(5,201)
  201 FORMAT(1H1)
      GO TO 2
   69 CALL EXIT
  100 FORMAT(33H ROOFTOP ANTENNA POINTING PROGRAM)
  101 FORMAT(33H SET ALL CONSOLE SWITCHES TO ZERO)
   50 FORMAT('    RTLOOP START ERROR',2X,I6)
      END
```

SUBROUTINE HINIT

```
      SUBROUTINE INITAL
C
C        THIS ROUTINE PERFORMS ALL NECESSARY CALCULATIONS AND SETS
C        ALL VARIABLES TO THE REQUIRED STATUS
C
      COMMON /AL/ALLAT,ALLNG,ALRAD,RAD,SLAT,SLNG,CLAT,CLNG
      COMMON /COORD/XSAT,YSAT,ZSAT,XAL,YAL,ZAL,XX,YY,ZZ,XXA,YYA,ZZA
      COMMON /FDUM1/ALAT,F2,F3,F4,F5,F6,F7,F8,F9,F10
      DATA ASMINT,STATE1,PNT1/6RASMINT,6RSTATE1,6RPNT1  /
C        CONVERT LONGITUDE AND LATITUDE TO RADIANS
   10 CALL DATAIO
      ALLAT=ALLAT/RAD
      A=ALAT/RAD
      ALAT=90.0-ALAT
      CLLAT=ALAT/RAD
      ALLNG=ALLNG/RAD
C        COMPUTE SINE AND COSINE OF LATITUDE AND LONGITUDE
      SLAT=SIN(A)
      CLAT=COS(A)
   50 FORMAT('  ASMINT START ERROR')
      SLNG=SIN(ALLNG)
      CLNG=COS(ALLNG)
C        COMPUTE LAB POSITION IN GEOCENTRIC COORDINATES
      XAL=CLAT*CLNG*ALRAD
      YAL=CLAT*SLNG*ALRAD
      ZAL=SLAT*ALRAD
      SLAT = SIN(CLLAT)
      CLAT = COS(CLLAT)
      RETURN
      END
```

SUBROUTINE HDATA

```
      SUBROUTINE DATAIO
C         IDAY - DAY OF THE YEAR STARTING FROM 1 JAN. - INTEGER
C         DTC - MIT TIME BIAS IN SECONDS - REAL
C         TBID - MIDTIME OF THE CURRENT PEROID IN HOURS - REAL
C         FREQ - CARRIER FREQUENCY IN HZ - REAL
C         II - 1 TO INITALIZE THE DISK FILE, 0 TO LEAVE AS IS - INTEGER
C         LES - NUMBER TO INDICATE SATELLITE UNDER USE (8 OR 9) - INTEGER
C     RH =RELATIVE HUMIDITY
C     PP =ABSOLUTE PRESSURE IN MILLIBARS
C     TFAR =TEMPERATURE IN DEGREES F
      DIMENSION IARAY(350)
      COMMON /C/XC(12),YC(12),ZC(12),XCA(12),YCA(12),ZCA(12)
      COMMON /C/XCB(12),YCB(12),ZCB(12),KK,IVAR
      COMMON /FL/K,L,M
      COMMON /SATC/XSATC,YSATC,ZSATC,DTC
      COMMON /BIAS/AZBIAS,ELBIAS,TIBIAS
      COMMON /TDDOP/FREQ,DOPLER,TDLAY
      COMMON /PTHN/PP,TC,TK,PW,AN
      COMMON /IDUM1/LES,I2,IEOF,I4,I5,IDAY,IERFLG,I8,I9,I10
      COMMON /AL/ALLAT,ALLNG,ALRAD,RAD,SLAT,SLNG,CLAT,CLNG
      COMMON /TT/TMID,TETIM,TF,TI,TETI3,DT
      COMMON /FDUM1/ALAT,F2,F3,F4,F5,F6,F7,F8,F9,F10
      K = 1
      IODAY=IDAY
      READ(1,101,END=14)IDAY,DTC,FREQ,II,LES
C         READ REFRACTION INPUTS AND PREPARE THEM
      READ(1,112,END=14)PP,TFAR,RH,AZBIAS,ELBIAS,ALLAT,ALLNG
      RLAT=ALLAT/RAD
      ALAT=ALLAT-.19243*SIN(2.*RLAT)+.00032314*SIN(4.*RLAT)
     1-.0000007222*SIN(6.*RLAT)
      ALRAD=6378160.*(.998327073+.001676438*COS(2.*RLAT)
     1-.000003519*COS(4.*RLAT)+.000000008*COS(6.*RLAT))
      PP=PP*33.8636
      TC=(TFAR-32.)*5./9.
      TK=TC+273.18
      AK1=77.6
      AK2=0.
      AK3=4810.
      TWV=TFAR+460.
      IF(TFAR.LT.32.0)GO TO 30
      PW=RH*1.53197*EXP(16.17916*(TWV-590.)/TWV)
      GO TO 40
   30 PW=RH*.061003*EXP(22.51093*(TWV-492.)/TWV)
   40 AN=AK1*PP/TK+AK2*PW/TK+AK3*PW/TK**2
      KK = 0
      IF(IDAY.EQ.IODAY)GO TO 15
    3 IFLG=0
      JFLG=0
    2 KK = KK + 1
    7 READ(3'KK,END=14)(IARAY(I),I=1,345)
      DECODE (679,106,IARAY) LLES,(XCA(I),I=1,10),(YCA(I),I=1,10),
     1    (ZCA(I),I=1,10),JDAY
      IF(JDAY-IDAY)8,9,8
    8 KK=KK+1
```

```
      GO TO 7
    9 KK = KK + 1
      IF(TMID.EQ.21600.)GO TO 5
      IF(JFLG.NE.0)GO TO 5
      JFLG=1
      GO TO 7
    5 IF(LES.EQ.8)GO TO 6
      IF(IFLG.NE.0)GO TO 6
      KK=KK+3
      IFLG=1
      JFLG=0
      GO TO 7
    6 READ(3'KK,END=14)(IARAY(I),I=1,345)
      DECODE (679,106,IARAY) LLES,(XCB(I),I=1,10),(YCB(I),I=1,10),
     1   (ZCB(I),I=1,10),JDAY
      J=JDAY-1
      DO 12 I=1,10
   12 CONTINUE
      DO 13 I=1,10
   13 CONTINUE
   15 IF(TMID.EQ.21600.)GO TO 10
      DO 11 I=1,10
      X=XCA(I)
      Y=YCA(I)
      Z=ZCA(I)
      XCA(I)=XCB(I)
      YCA(I)=YCB(I)
      ZCA(I)=ZCB(I)
      XCB(I)=X
      YCB(I)=Y
   11 ZCB(I)=Z
   10 RETURN
C         SET END OF FILE FLAG
   14 IEOF=1
      RETURN
  106 FORMAT (10XI1,7XE16.10,29(6XE16.10),4XI3)
  101 FORMAT(I3,3XE14.7,3XE14.7,3XI1,3XI1)
  112 FORMAT(7F10.4)
      END
```

SUBROUTINE HANTEN

```
      SUBROUTINE ANTENA
C         THIS SUBROUTINE COMPUTES SATELLITE POSITION AND DESIRED
C         ANTENNA AZIMUTH AND ELEVATION
      COMMON /TT/TMID,TETIM,TF,TI,TETI3,DT
      COMMON /AL/ALLAT,ALLNG,ALRAD,RAD,SLAT,SLNG,CLAT,CLNG
      COMMON /C/XC(12),YC(12),ZC(12),XCA(12),YCA(12),ZCA(12)
      COMMON /C/XCB(12),YCB(12),ZCB(12),KK,IVAR
      COMMON /COORD/XSAT,YSAT,ZSAT,XAL,YAL,ZAL,XX,YY,ZZ,XXA,YYA,ZZA
      COMMON /ANT/AZI,ELEV,AZIIN,ELEIN,SDBIT,DABIT,ERA,ERE
      COMMON /BIAS/AZBIAS,ELBIAS,TIBIAS
      COMMON /SATC/XSATC,YSATC,ZSATC,DTC
C         READ TIME FROM THE CLOCK
      CALL CTIME
C         COMPUTE SATELLITE POSITION
      XSAT=0.0
      YSAT=0.0
      ZSAT=0.0
      DO 1 I=1,9
      J=11-I
      XSAT=(XSAT+XC(J))*TETI3
      YSAT=(YSAT+YC(J))*TETI3
    1 ZSAT=(ZSAT+ZC(J))*TETI3
      XSAT=XSAT+XC(1)+XSATC
      YSAT=YSAT+YC(1)+YSATC
      ZSAT=ZSAT+ZC(1)+ZSATC
C         COMPUTE VECTOR FROM AVIONICS LAB TO SATELLITE
      XX=XSAT-XAL
      YY=YSAT-YAL
      ZZ=ZSAT-ZAL
C         ROTATE VECTOR INTO AVIONICS LAB COORDINATES
      XXA=CLAT*CLNG*XX+CLAT*SLNG*YY-SLAT*ZZ
      YYA=-SLNG*XX+CLNG*YY
      ZZA=SLAT*CLNG*XX+SLAT*SLNG*YY+CLAT*ZZ
C         COMPUTE AZIMUTH AND ELEVATION FOR ANTENNA
      AXXA=ABS(XXA)
      AYYA=ABS(YYA)
      AZZA=ABS(ZZA)
      AXYA=SQRT(XXA*XXA+YYA*YYA)
      AZI=ATAN2(AYYA,AXXA)
      ELEV=ATAN2(AZZA,AXYA)
C         PUT AZIMUTH INTO CORRECT QUADRANT
      IF(XXA)10,12,12
   10 IF (YYA)13,14,15
   13 AZI=(360./RAD)-AZI
      GO TO 25
   14 AZI=0.0
      GO TO 25
   15 AZI=AZI
      GO TO 25
   12 IF(YYA)16,17,18
   16 AZI=(180./RAD)+AZI
      GO TO 25
   17 AZI=180./RAD
      GO TO 25
   18 AZI=(180./RAD)-AZI
```

SUBROUTINE HANTEN (CONTD)

```
25 AZI=AZI*RAD
   ELEV=ELEV*RAD
   IF(ZZA.LT.0.0)ELEV=-ELEV
   CALL REFCT
   RETURN
   END
```

SUBROUTINE HTIME

```
      SUBROUTINE CTIME
C         THIS ROUTINE PROVIDES TIME FOR THE SATELLITE POSITION CALCULATION
      COMMON /TT/TMID,TETIM,TF,TI,TETI3,DT
      COMMON /BIAS/AZBIAS,ELBIAS,TIBIAS
      COMMON /C/XC(12),YC(12),ZC(12),XCA(12),YCA(12),ZCA(12)
      COMMON /C/XCB(12),YCB(12),ZCB(12),KK,IVAR
      IF(TETIM.LT.43200.)GO TO 4
      DO 10 I=1,10
      XC(I)=XCB(I)
      YC(I)=YCB(I)
   10 ZC(I)=ZCB(I)
      TMID=64800.
      TI=43200.
      TF=86400.
      TETI3=(TETIM+DT+TIBIAS-TMID)/21600.
      RETURN
    4 DO 20 I=1,10
      XC(I)=XCA(I)
      YC(I)=YCA(I)
   20 ZC(I)=ZCA(I)
      TMID=21600.
      TI=0.0
      TF=43200.
      TETI3=(TETIM+DT+TIBIAS-TMID)/21600.
      RETURN
      END
```

```
C          ALASKA ANTENNA POINTING PROGRAM
C
C          THIS TASK IS INSTALLED UNDER THE RSX-11D OPERATING SYSTEM.
C          ALASKA GENERATES LOOK ANGLES TO LES 8 OR LES 9 FOR
C          A GIVEN LAT, LNG, AND TIME.
C
C          INPUT DATA HAS UNITS OF DEGREES, METERS, AND SECONDS
C
      COMMON /TT/TMID,TETIM,TF,TI,TETI3,DT
      COMMON /AL/ALLAT,ALLNG,ALRAD,RAD,SLAT,SLNG,CLAT,CLNG
      COMMON /C/XC(12),YC(12),ZC(12),XCA(12),YCA(12),ZCA(12)
      COMMON /C/XCB(12),YCB(12),ZCB(12),KK,IVAR
      COMMON /COORD/XSAT,YSAT,ZSAT,XAL,YAL,ZAL,XX,YY,ZZ,XXA,YYA,ZZA
      COMMON /ANT/AZI,ELEV,AZIIN,ELEIN,SDBIT,DABIT,ERA,ERE
      COMMON /FL/K,L,M
      COMMON /BIAS/AZBIAS,ELBIAS,TIBIAS
      COMMON /SATC/XSATC,YSATC,ZSATC,DTC
      COMMON /DOTI/XDOTI,YDOTI,ZDOTI,XDOT,YDOT
      COMMON /TDOOP/FREQ,DOPLER,TDLAY
      COMMON /PTHN/PP,TC,TK,PW,AN
      COMMON /ANTC/IOUNT,INTFLG
      COMMON /IDUM1/LES,N,IEOF,IAMFLG,IPMFLG,IDAY,IERFLG,I8,I9,I10
      DATA RTLOOP/6RRTLOOP/
      DATA RAPP,STATE1,PNT1,DOIT/6RRAPP  ,6RSTATE1,6RPNT1  ,6RDOIT  /
C          PROGRAM NAME AND INSTRUCTIONS

      CALL ASSIGN(3,'DK0:DAVEM.DAT;1',15)
      DEFINE FILE 3(1000,345,U,IVAR)
C          INITIALIZATION ROUTINE
      IDAY=0
      IEOF=0
      IAMFLG=0
      IPMFLG=0
      IERFLG=0
      TMID=0.
      TETIM=0.
      TF=0.
      TI=0.
      TETI3=0.
      DT=2.5
      ALLAT=39.5882
      ALLNG=-84.0829
      ALRAD=6369762.
      RAD=57.29578
      DO 1 I=1,12
      XC(I)=0.
      YC(I)=0.
      ZC(I)=0.
      XCA(I)=0.
      YCA(I)=0.
      ZCA(I)=0.
      XCB(I)=0.
      YCB(I)=0.
    1 ZCB(I)=0.
```

```
      KK=0
      IVAR=0
      XSAT=0.
      YSAT=0.
      ZSAT=0.
      XAL=0.
      YAL=0.
      ZAL=0.
      XX=0.
      YY=0.
      ZZ=0.
      XXA=0.
      YYA=0.
      ZZA=0.
      AZI=0.
      ELEV=0.
      AZIIN=0.
      ELEIN=0.
      SDBIT=0.549325E-02
      DABIT=0.03125
      ERA=0.
      ERE=0.
      SLAT=0.
      SLNG=0.
      CLAT=0.
      CLNG=0.
      K=0
      L=0
      M=0
      AZBIAS=0.
      ELBIAS=0.
      TIBIAS=0.
      XSATC=0.
      YSATC=0.
      ZSATC=0.
      DTC=0.
      XDOTI=0.
      YDOTI=0.
      ZDOTI=0.
      XDOT=0.
      YDOT=0.
      FREQ=0.
      DOPLER=0.
      TDLAY=0.
      PP=0.
      TC=0.
      TK=0.
      PW=0.
      AN=0.
      IOUNT=1
      INTFLG=0
    2 TMID=21600.
      CALL INITAL
      IF(IEOF.EQ.0)GO TO 3
```

```
      WRITE(6,52)
      GO TO 69
   52 FORMAT('    CAN NOT LOCATE EPHEMERIS DATA. RAPP CANCELLED.')
C
    3 ALAT=ALLAT*RAD
      ALNG=ALLNG*RAD
      WRITE(6,201)ALAT
  201 FORMAT(1H ,'ENTER INITIAL TIME FOR LATITUDE ',F5.1)
      READ(6,102)ITIME,JTIME
  102 FORMAT(2I2)
      DO 30 I=1,4
      TETIM=3600.*ITIME+60.*JTIME
      CALL ANTENA
      WRITE(5,200)IDAY,LES,ITIME,JTIME,ALAT,ALNG,AZI,ELEV
  200 FORMAT(1H ,I4,2X,I2,2X,2I2,2X,F6.1,2X,F6.1,2X,F7.2,2X,F7.2)
   30 ITIME=ITIME+1
      GO TO 2
   69 CALL EXIT
   50 FORMAT('    RTLOOP START ERROR',2X,I6)
      END
```

TASK WRITER

```
      COMMON /TT/TMID,TETIM,TF,TI,TETI3,DT
      COMMON /ANT/AZI,ELEV,AZIIN,ELEIN,SDBIT,DABIT,ERA,ERE
      HOUR=TETIM/3600.
      IHOUR=HOUR
      AMIN=(HOUR-IHOUR)*60.
      MIN=AMIN
      SEC=(AMIN-MIN)*60.
      ISEC=SEC
      WRITE(5,200)AZI,AZIIN,ERA,ELEV,ELEIN,ERE,IHOUR,MIN,ISEC
  200 FORMAT(1H ,3(F8.3,2X),'*',2X,3(F8.3,2X),'*',2X,2(I2,':'),I2)
      CALL EXIT
      END
```

```
      IMPLICIT DOUBLE PRECISION (A-H),(O-Z)
      DIMENSION IARAY(350),XC(12),YC(12),ZC(12),XCA(12),YCA(12),ZCA(12),
     1XCB(12),YCB(12),ZCB(12)
      KK = 0
    3 CALL ASSIGN(3,'DK0:DAVEM.DAT;1',15)
      LES=8
      IFLG=0
      JFLG=0
      DEFINE FILE 3(1000,345,U,IVAR)
    2 KK = KK + 1
    7 READ(3'KK,END=14)(IARAY(I),I=1,345)
      DECODE(686,106,IARAY)LLES,(XCA(I),I=1,10),(YCA(I),I=1,10),
     1(ZCA(I),I=1,10),JDAY,JT1,JT2
      IF(LES.NE.LLES)WRITE(5,101)
      WRITE(5,105)LLES
  101 FORMAT(1H1)
      LES=LLES
      WRITE(5,102)JDAY
      WRITE(5,107)
      DO 10 I=1,10
   10 WRITE(5,109)XCA(I),YCA(I),ZCA(I)
      WRITE(5,108)JT1,JT2
  108 FORMAT(1H ,'TIME:',A3,A4//)
      KK=KK+1
    6 READ(3'KK,END=14)(IARAY(I),I=1,345)
      DECODE(686,106,IARAY)LLES,(XCB(I),I=1,10),(YCB(I),I=1,10),
     1(ZCB(I),I=1,10),JDAY,JT1,JT2
      WRITE(5,105)LLES
      WRITE(5,102)JDAY
      WRITE(5,107)
      DO 20 I=1,10
   20 WRITE(5,109)XCB(I),YCB(I),ZCB(I)
      WRITE(5,110)JT1,JT2
  110 FORMAT(1H ,'TIME:',A3,A4/////)
      GO TO 2
   14 IEOF=1
      CALL EXIT
  102 FORMAT(12H DAY NUMBER I3/)
  105 FORMAT (15H SATELLITE LES-I1)
  106 FORMAT(10X,I1,7XE16.10,29(6XE16.10),4XI3,A3,A4)
  107 FORMAT (17H    X COEFFICIENTS,11X14HY COEFFICIENTS,
     1  11X14HZ COEFFICIENTS)
  109 FORMAT(2XE17.10,8X,E17.10,8X,E17.10)
  111 FORMAT(30X,'P.M. EPHEMERIS FOR DAY ',I4)
      END
```

AD-A047 999 AIR FORCE AVIONICS LAB WRIGHT-PATTERSON AFB OHIO F/G 17/2

ROOFTOP ANTENNA POINTING PROGRAM. (U)

NOV 77 P F HUMEL

UNCLASSIFIED AFAL-TR-77-228 NL

2 of 2

ADA047 999

END

DATE

FILMED

1 -78

DDC

TASK BLODAT

```
      BLOCK DATA
C        THIS ROUTINE INITIALIZES ALL COMMON BLOCKS
      COMMON /TT/TMID,TETIM,TF,TI,TETI3,DT
      COMMON /AL/ALLAT,ALLNG,ALRAD,RAD,SLAT,SLNG,CLAT,CLNG
      COMMON /C/XC(12),YC(12),ZC(12),XCA(12),YCA(12),ZCA(12)
      COMMON /C/XCB(12),YCB(12),ZCB(12),KK,IVAR
      COMMON /COORD/XSAT,YSAT,ZSAT,XAL,YAL,ZAL,XX,YY,ZZ,XXA,YYA,ZZA
      COMMON /ANT/AZI,ELEV,AZIIN,ELEIN,SDBIT,DABIT,ERA,ERE
      COMMON /FL/K,L,M
      COMMON /BIAS/AZBIAS,ELBIAS,TIBIAS
      COMMON /SATC/XSATC,YSATC,ZSATC,DTC
      COMMON /DOTI/XDOTI,YDOTI,ZDOTI,XDOT,YDOT
      COMMON /TDDOP/FREQ,DOPLER,TDLAY
      COMMON /PTHN/PP,TC,TK,PW,AN
      COMMON /ANTC/IOUNT,INTFLG
        COMMON /BBIAS/ABIAS,EBIAS
        COMMON /IDUM1/I1,I2,I3,I4,I5,I6,I7,I8,I9,I10
        COMMON /IDUM2/I11,I12,I13,I14,I15,I16,I17,I18,I19,I20
        COMMON /FDUM1/F1,F2,F3,F4,F5,F6,F7,F8,F9,F10
        COMMON /FDUM2/F11,F12,F13,F14,F15,F16,F17,F18,F19,F20
      DATA TMID,TETIM,TF,TI,TETI3,DT/5*0.0,2.5/
      DATA ALLAT,ALLNG,ALRAD,RAD/-39.5882,-84.0829,6369762.,57.29578/
      DATA XC,YC,ZC,XCA,YCA,ZCA,XCB,YCB,ZCB,KK,IVAR/108*0.0,0,0/
      DATA XSAT,YSAT,ZSAT,XAL,YAL,ZAL,XX,YY,ZZ,XXA,YYA,ZZA/12*0.0/
      DATA AZI,ELEV,AZIIN,ELEIN,SDBIT,DABIT/4*0.0,.549325E-02,0.03125/
      DATA ERA,ERE,SLAT,SLNG,CLAT,CLNG/6*0.0/
      DATA K,L,M,AZBIAS,ELBIAS,TIBIAS/0,0,0,3*0.0/
      DATA XSATC,YSATC,ZSATC,DTC/4*0.0/
      DATA XDOTI,YDOTI,ZDOTI,XDOT,YDOT/5*0.0/
      DATA FREQ,DOPLER,TDLAY/3*0.0/
      DATA PP,TC,TK,PW,AN/5*0.0/
      DATA IOUNT,INTFLG/0,0/
        DATA ABIAS,EBIAS/2*0.0/
        DATA I1,I2,I3,I4,I5,I6,I7,I8,I9,I10/10*0/
        DATA I11,I12,I13,I14,I15,I16,I17,I18,I19,I20/10*0/
        DATA F1,F2,F3,F4,F5,F6,F7,F8,F9,F10/10*0.0/
        DATA F11,F12,F13,F14,F15,F16,F17,F18,F19,F20/10*0.0/
      END
```

PROGRAM PAUL

```
        .TITLE  TAPE PRINT
        .PSECT  FTNCOM,GBL,OVR
        FTNBUF=.+0
        .=.+1380.
        .PSECT  MAINPROG
        .MCALL  .INIT,.TRAN,.WRITE,.WAIT,.EXIT,.RLSE
        .MCALL  .OPEN,.CLOSE
;
BEGIN:
        CLR     R2
        .INIT   #TAPE
        .INIT   #PRNT
        .OPEN   #PRNT,#PRNTFB
;
ERECD:
;
        MOV     #690.,TCNT          ;READ DETAIL REC
        .TRAN   #TAPE,#TAPETB
        .WAIT   #TAPE
;
        BIT     #40000,@#172520 ;MAGTAPE CSR TO CHECK FOR EOF
        BEQ     1$                  ;TEST IF SECOND EOF ENCOUNTERED
        TST     R2                  ;TEST IF SECOND EOF ENCOUNTERED
        BNE     EFILE               ;IF TRUE, CLOSE FILES
        INC     R2
1$:     CMP     BUFR+4,#"EP
        BNE     ERECD
        MOV     #FTNBUF,R0
        MOV     #BUFR,R1
        MOV     #-690.,FTNCNT
CONT:
        MOV     (R1)+,(R0)+         ;MOVE TO FORTRAN BUFFER
        INC     FTNCNT
        BNE     CONT

;
        MOV     #690.,CNT           ;WRITE
        .WRITE  #PRNT,#PRNTBF
        .WAIT   #PRNT
;
        BR      ERECD
;
EFILE:
;
        .CLOSE  #PRNT
        .RLSE   #TAPE
        .RLSE   #PRNT
;
1$:     BR      2$
2$:     BR      1$
;
NGRPS: .WORD -101.
NRECS:  .WORD   -21.
FTNCNT: .WORD   0
```

```
RCNT:   .WORD   0
;
;TAPE LINKBLOCK
        .WORD   0
TAPE:   .WORD   0
        .RAD50  ^TAP^
        .WORD   1
        .RAD50  ^MT^
;
;PRINTER LINKBLOCK
        .WORD   0
PRNT:   .WORD   0
        .RAD50  ^PRT^
        .WORD   1
        .RAD50  ^DK^
;
;TAPE TRAN BLOCK
TAPETB: .WORD   0
        .WORD   BUFR
TCNT:   .WORD   0
        .WORD   4
        .WORD   0
;
;PRINTER BUFFER
PRNTBF: .WORD   1380.
        .WORD   0
CNT:    .WORD   0
BUFR:   .BLKW   690.
;
;PRINTER FILEBLOCK
        .WORD   0
        .WORD   2
PRNTFB: .RAD50  ^DAV^
        .RAD50  ^EM^
        .RAD50  ^DAT^
        .WORD   0
        .WORD   0
        .END    BEGIN
```

PROGRAM RCA

```
      IMPLICIT DOUBLE PRECISION (A-H),(O-Z)
      DIMENSION IARAY(350),X(30),NEL(420),NUM(1000)
C         SET UP
      CALL ASSIGN(3,'DK0:DAVEM.DAT',13,J)
      DEFINE FILE 3(1000,345,U,IVAR)
      CALL ASSIGN (4,'DK0:RCA.LST',11,J)
      WRITE(6,202)
  202 FORMAT(1H ,' ENTER DAY OF YEAR AS DDD')
      READ(6,100)IDAY
  100 FORMAT(I3)
      KOUNT=1
      KK=1
C         READ EPHEMERIS DATA
    1 READ(3'KK,END=14)(IARAY(I),I=1,345)
      DECODE(682,101,IARAY)LES,(X(N),N=1,30),JDAY,JTIME
  101 FORMAT(10X,I1,7X,E16.10,29(6X,E16.10),4X,I3,1X,I2)
      IF(JDAY-IDAY)2,3,2
    2 KK=KK+1
      GO TO 1
    3 KK=KK+1
C         SPOOL DATA ONTO DISK
      WRITE(4,200)KOUNT,(X(N),N=1,30),JTIME
  200 FORMAT('ORBIT ',I1,' ',6(5(G17.10,' ')/),I2,':00:00.000')
      KOUNT=KOUNT+1
      IF(KOUNT.LT.5)GO TO 1
      GO TO 35
   34 WRITE(6,201)
  201 FORMAT(1H ,'COULD NOT LOCATE EPHEMERIS DATA ON DISK')
   35 KOUNT=1
      LTR=1
C         READ DATA OFF DISK
    4 READ(4,102)(NEL(I),I=1,73)
  102 FORMAT(8A1,1X,4(13A1,5X),13A1)
      READ(4,103)(NEL(I),I=74,138)
  103 FORMAT(1X,4(13A1,5X)13A1)
      READ(4,103)(NEL(I),I=139,203)
      READ(4,103)(NEL(I),I=204,268)
      READ(4,103)(NEL(I),I=269,333)
      READ(4,103)(NEL(I),I=334,398)
      READ(4,104)(NEL(I),I=399,410)
  104 FORMA[illegible]A1)
C         SE[illegible]
      IF(NE[illegible].NE.'1')NEL(399)='0'
      I=1
      J=0
      NUM(1)='6'
      IF(LTR.EQ.1)NUM(1)='?'
      IF((LTR.EQ.1).AND.(KOUNT.GT.1))I=0
C         CONVERT ASCII TO BAUD-O
    5 I=I+1
      J=J+1
      IF(J.GT.410)GO TO 25
      N=NEL(J)
      IF(N.EQ.'O')GO TO 6
```

```
      IF(N.EQ.'R')GO TO 7
      IF(N.EQ.'B')GO TO 8
      IF(N.EQ.'I')GO TO 9
      IF(N.EQ.'T')GO TO 10
      IF(N.EQ.'-')GO TO 11
      IF(N.EQ.'.')GO TO 12
      IF(N.EQ.'0')GO TO 13
      IF(N.EQ.'1')GO TO 14
      IF(N.EQ.'2')GO TO 15
      IF(N.EQ.'3')GO TO 16
      IF(N.EQ.'4')GO TO 17
      IF(N.EQ.'5')GO TO 18
      IF(N.EQ.'6')GO TO 19
      IF(N.EQ.'7')GO TO 20
      IF(N.EQ.'8')GO TO 21
      IF(N.EQ.'9')GO TO 22
      IF(N.EQ.' ')GO TO 23
      IF(N.EQ.':')GO TO 24
      WRITE(6,203)N
  203 FORMAT(1H ,'ERROR- INVALID CHARACTER IN INPUT FILE ',A4)
      NUM(I)='X'
      I=I+1
      GO TO 5
    6 IF(LTR.EQ.1)GO TO 66
      NUM(I)='?'
      LTR=1
      I=I+1
   66 NUM(I)='0'
      GO TO 5
    7 IF(LTR.EQ.1)GO TO 77
      NUM(I)='?'
      LTR=1
      I=I+1
   77 NUM(I)='T'
      GO TO 5
    8 IF(LTR.EQ.1)GO TO 88
      NUM(I)='?'
      LTR=1
      I=I+1
   88 NUM(I)='2'
      GO TO 5
    9 IF(LTR.EQ.1)GO TO 99
      NUM(I)='?'
      LTR=1
      I=I+1
   99 NUM(I)='L'
      GO TO 5
   10 IF(LTR.EQ.1)GO TO 1010
      NUM(I)='?'
      LTR=1
      I=I+1
 1010 NUM(I)='!'
      GO TO 5
   11 IF(LTR.EQ.0)GO TO 1111
```

```
      NUM(I)='6'
      LTR=0
      I=I+1
 1111 NUM(I)='F'
      GO TO 5
   12 IF(LTR.EQ.0)GO TO 1212
      NUM(I)='6'
      LTR=0
      I=I+1
 1212 NUM(I)='8'
      GO TO 5
   13 IF(LTR.EQ.0)GO TO 1313
      NUM(I)='6'
      LTR=0
      I=I+1
 1313 NUM(I)='-'
      GO TO 5
   14 IF(LTR.EQ.0)GO TO 1414
      NUM(I)='6'
      LTR=0
      I=I+1
 1414 NUM(I)='/'
      GO TO 5
   15 IF(LTR.EQ.0)GO TO 1515
      NUM(I)='6'
      LTR=0
      I=I+1
 1515 NUM(I)='&'
      GO TO 5
   16 IF(LTR.EQ.0)GO TO 1616
      NUM(I)='6'
      LTR=0
      I=I+1
 1616 NUM(I)='B'
      GO TO 5
   17 IF(LTR.EQ.0)GO TO 1717
      NUM(I)='6'
      LTR=0
      I=I+1
 1717 NUM(I)='T'
      GO TO 5
   18 IF(LTR.EQ.0)GO TO 1818
      NUM(I)='6'
      LTR=0
      I=I+1
 1818 NUM(I)='1'
      GO TO 5
   19 IF(LTR.EQ.0)GO TO 1919
      NUM(I)='6'
      LTR=0
      I=I+1
 1919 NUM(I)='*'
      GO TO 5
   20 IF(LTR.EQ.0)GO TO 2020
```

```
      NUM(I)='6'
      LTR=0
      I=I+1
 2020 NUM(I)='N'
      GO TO 5
   21 IF(LTR.EQ.0)GO TO 2121
      NUM(I)='6'
      LTR=0
      I=I+1
 2121 NUM(I)='L'
      GO TO 5
   22 IF(LTR.EQ.0)GO TO 2222
      NUM(I)='6'
      LTR=0
      I=I+1
 2222 NUM(I)='0'
      GO TO 5
   23 IF(LTR.EQ.1)GO TO 2323
      NUM(I)='?'
      LTR=1
      I=I+1
 2323 NUM(I)='H'
      GO TO 5
   24 IF(LTR.EQ.0)GO TO 2424
      NUM(I)='6'
      LTR=0
      I=I+1
 2424 NUM(I)='1'
      GO TO 5
   25 IF(LTR.EQ.1)GO TO 2525
      NUM(I)='?'
      LTR=1
      I=I+1
 2525 NUM(I)='P'
      DO 30 J=1,25
      I=I+1
   30 NUM(I)='H'
C        OUTPUT TO PAPER TAPE
      CALL IO(I,NUM(1))
      KOUNT=KOUNT+1
      IF(KOUNT.LT.5)GO TO 4
      CALL EXIT
      END
```

SUBROUTINE IO

```
.TITLE    IO
;         THIS PROGRAM PERFORMS A DIRECT, UNFORMATTED OUTPUT TO PAPER TAPE
.GLOBL    IO,I,NUM
          R0=%0
          R1=%1
          R2=%2
          R5=%5
          SP=%6
I:        .WORD     0
NUM:      .BLKW     1000.
PSR:      .WORD     177564    ; 177554 FOR HIGH SPEED PUNCH
PDB:      .WORD     177566    ; 177556 FOR HIGH SPEED PUNCH
IO:       MOV       R0,-(SP)
          MOV       R1,-(SP)
          MOV       R2,-(SP)
          MOV       2(R5),R0
          MOV       (R0),I
          MOV       4(R5),R0
          MOV       #NUM,R1
          MOV       I,R2
LOOP:     MOV       (R0)+,(R1)+
          DEC       R2
          BNE       LOOP
          MOV       I,R0
          MOV       #NUM,R1
PUNCH:    TSTB      @PSR
          BPL       PUNCH
          MOVB      (R1)+,@PDB
          TSTB      (R1)+
          DEC       R0
          BNE       PUNCH
          MOV       (SP)+,R2
          MOV       (SP)+,R1
          MOV       (SP)+,R0
          RTS       R5
          .END
```

PROGRAM TRW

```
      DIMENSION IARAY(682),M(840)
      CALL ASSIGN(3,'DK0:DAVEM.DAT',13,J)
      DEFINE FILE 3(1000,345,U,IVAR)
      READ(6,100)IDAY
  100 FORMAT(I3)
      KOUNT=1
      KK=1
    1 READ(3'KK,END=14)(IARAY(I),I=1,345)
      DECODE(682,102,IARAY)LES,(M(N),N=1,30),JDAY,JTIME
  102 FORMAT(10X,I1,7X,E16.10,29(6X,E16.10),4X,I3,1X,I2)
      KK=KK+1
      IF(JDAY-IDAY)1,3,1
    3 DECODE(682,101,IARAY)(M(N),N=1,682)
  101 FORMAT(682A1)
      GO TO 4
   14 WRITE(6,201)
  201 FORMAT(1H ,'COULD NOT LOCATE EPHEMERIS DATA ON DISK')
      GO TO 1000
    4 I=836
      M(683)='0'
      M(684)='0'
      M(685)='0'
      M(686)='0'
      DO 10 N=687,836
   10 M(N)=''
      CALL IO(I,M(1))
      KOUNT=KOUNT+1
      IF(KOUNT.LT.5)GO TO 1
 1000 CALL EXIT
      END
```

PROGRAM SATSPT

```
      INTEGER OUTOPT,PLTFLG,PLTPTS,SENSE,SALAT,SSLAT,SVLAT,
     1SALONG,SSLONG,SVLONG,DECON
      DIMENSION ZETA(20),REMARK(7),PLTLAT(362),PLTLNG(362)
      MI=1
      MO=5
    1 READ(MI,110,END=1000)OUTOPT,ZETA
      IF (OUTOPT.LE.3) GO TO 2
      WRITE (MO,222)
      GO TO 1000
    2 J=1
      NELEV=0
 3    I=1
      PLTPTS=0
      ITER=9999
      IF (J.GT.1) ITER=ZITER+.1
 5    N=0
C  IF MORE LONGITUDE ITERATIONS, BRANCH AHEAD ELSE PLOT ISO-CURVE
      IF(I .GT. ITER) GO TO 11
111   WRITE(MO,200)ZETA
      IF (OUTOPT.GE.2)  WRITE (MO,205)
      WRITE (MO,207)
      WRITE(MO,210)
      WRITE(MO,220)
 10   IF(I.GT.ITER) GO TO 11
      IF (I.GT.1)  GO TO 22
11    IF(OUTOPT .EQ. 3 .AND. PLTFLG .NE. 0 .AND. I .NE. 1)
     XCALL PLTISO(PLTLNG,PLTLAT,PLTPTS)
       IF(J .LE. NELEV) GO TO 13
      READ(MI,120)NHOUR,NMIN,NSEC,NPDF,ALAT,ALONG,SLAT,SLONG,H,HEAD,
     XRAD,DME,VARMAG,PITCH,PLTFLG,REMARK
      IF (NPDF.LE.2) GO TO 12
      WRITE (MO,223)
      GO TO 1000
 12   IF (NHOUR.NE.99) GO TO 15
      IF (N.EQ.0) GO TO 1
      WRITE  (MO,270) ZETA
      IF (OUTOPT.GE.2) WRITE(MO,205)
      GO TO 1
 13   IF (I.EQ.1) GO TO 15
      THETAT=THETAT+ELSTEP
      IF (THETAT.LE.90.0) GO TO 14
      WRITE (MO,225)
      GO TO 1000
 14   J=J+1
      IF (N.NE.0)  WRITE (MO,270) ZETA
      IF (N.NE.0)  WRITE (MO,205)
      IF (J.GT.NELEV) GO TO 5
      GO TO 3
15    VLAT=0.0
      VLONG=0.0
      IF (OUTOPT.GE.2) GO TO 16
      Z1=ABS(ALAT)
      IF (Z1.LE.90.0) GO TO 16
      WRITE (MO,224)
```

```
      GO TO 1000
   16 Z1=ABS(SLAT)
      IF (Z1.LE.90.0) GO TO 17
      WRITE (MO,224)
      GO TO 1000
 17   IF (J.EQ.1) GO TO 18
      ALONG=ALHOLD
      GO TO 22
 18   IF(NPDF.EQ.0) GO TO 20
      ALAT=CONDE(ALAT)
      ALONG=CONDE(ALONG)
      SLAT=CONDE(SLAT)
      SLONG=CONDE(SLONG)
      IF(NPDF.EQ.1) GO TO 20
      IF (OUTOPT.LT.2) GO TO 19
      WRITE (MO,221)
      GO TO 1000
 19   VLAT=ALAT
      VLONG=ALONG
      TRUE=RAD-VARMAG
      IF (TRUE.GT.360.0) TRUE=TRUE-360.0
      IF(TRUE.LT.0.0) TRUE=TRUE+360.0
      ALAT=VLAT+DME*COSD(TRUE)/60.0
      ALONG=VLONG-DME*SIND(TRUE)/COSD((ALAT+VLAT)/2.0)/60.0
      IF(ALONG.GT. 180.0) ALONG=ALONG-360.0
      IF(ALONG.LT.-180.0) ALONG=360.0+ALONG
C SET STAGE FOR PLOT CURVES TO COME
 20   IF (OUTOPT.LT.2) GO TO 21
      IF(OUTOPT .EQ. 3 .AND. PLTFLG .NE. 0)CALL PLTSET(SLONG,SLAT)
      STEP=ALAT
      ALAT=0.0
      THETAT=PITCH
      PITCH=0.0
      ZITER=HEAD
      ITER=ZITER+.1
      HEAD=0.0
      ELSTEP=VARMAG
      VARMAG=0.0
      IF (OUTOPT.NE.3) GO TO 22
      NELEV= DME+.1
      DME=0.0
      ALHOLD=ALONG
      GO TO 22
 21   B=90.0-ALAT
 22   A=90.0-SLAT
      HOLD=0.0
      NFLAG=0
      THETA=THETAT
      DELTA=ALONG-SLONG
C     IF(DELTA.LT.   0.0) DELTA=360.0+DELTA      STATEMENT DELETED
      IF (OUTOPT.LT.2) GO TO 41
      IF (H.GE.0.0) GO TO 23
      WRITE (MO,226)
      GO TO 1000
```

```
   23 IF(DELTA.LT.0.0)DLTA=360.0+DELTA
      APEX =ASIND(3441.1*COSD(THETA)/(H+3441.1))
      GAMMA=90.0-APEX-THETA
      TEST=SIND(A)*SIND( DLTA)/SIND(GAMMA)
      IF (TEST.GT.1.0) GO TO 65
      IF (TEST.LT.-1.0) GO TO 65
      ALPHA=ASIND(TEST)
      IF (A.GE.90.0) ALPHA= 180.0-ALPHA
      IF ( DLTA.LT.-.01)  GO TO 36
      IF ( DLTA.GT..01)  GO TO 34
      IF (A.GE.90.0) B=A-GAMMA
      IF (A.LT.90.0) B=A+GAMMA
      GO TO 38
 34   IF ( DLTA.LT.179.99)  GO TO 36
      IF ( DLTA.GT.180.01)  GO TO 36
      B=GAMMA-A
      IF (A.LE.90.0) GO TO 38
      B=GAMMA+A
      IF (B.LE.180.0)  B=0.0-B
      IF (B.GT.180.0) B=360.0-B
      GO TO 38
 36   B=2.0*ATAND(SIND(.5*( DLTA+ALPHA))*TAND(.5*(GAMMA-A))/SIND(.5*
     1( DLTA-ALPHA)))
 38   IF (GAMMA.LT.A) NFLAG=1
      IF((180.0-A).LT.GAMMA)NFLAG=0
      IF (B.GE.0.0) GO TO 40
      IF (A.EQ.90.0) GO TO 65
      B=0.0-B
      HOLD=ALONG
      ALONG=ALONG+180.0
      IF (ALONG.GT.180.0)  ALONG=ALONG-360.0
      DELTA=ALONG-SLONG
C     IF(DELTA.LT.   0.0) DELTA=360.0+DELTA      STATEMENT DELETED
      NFLAG=2
 40   ALAT=90.0-B
C ***********************PROGRAM CHANGE**********************************
   41 CALL AZI(A,B,ALAT,SLAT,DELTA,ALPHA)
C ***********************PROGRAM CHANGE**********************************
 49   GAMMA=ACOSD(COSD(A)*COSD(B)+SIND(A)*SIND(B)*COSD(DELTA))
      GCD=60.0*GAMMA
      SLANT= SQRT(H**2.0+(1.0-COSD(GAMMA))*(H+3441.1)*6882.2)
      THETA=ACOSD((H+3441.1)*SIND(GAMMA)/SLANT)
      IF ((COSD(GAMMA)).LT.(3441.1/(H+3441.1))) THETA=-THETA
      WPAFB=60.0*ACOSD(.64033*COSD(B)+.76810*SIND(B)*COSD( ABS(ALONG-84
     105)))
      TH=HEAD-VARMAG
      IF(TH.GT.360.0) TH=TH-360.0
      IF(TH.LT.0.0) TH=TH+360.0
      RBEAR=ALPHA-TH
      IF(RBEAR.GT.360.0) RBEAR=RBEAR-360.0
      IF(RBEAR.LT.0.0) RBEAR=RBEAR+360.0
      IF(VARMAG) 55,56,57
 55   SENSE=2H E
      GO TO 60
```

```
 56    SENSE=1H
       GO TO 60
 57    SENSE=2H W
   60 VARMAG= ABS(VARMAG)
       THETAC=THETA-ATAND(TAND(PITCH)*COSD(RBEAR))
       IF(OUTOPT.EQ.0) GO TO 70
C SAVE PLOT DATA IN DECIMAL FORM
       PLTPTS=PLTPTS + 1
       PLTLNG(PLTPTS)=XINCHS(ALONG,ALAT,YINCHS)
       PLTLAT(PLTPTS)=YINCHS
C  PLOT, BUT NO PRINT? YES, BRANCH AROUND PRINT
       IF(PLTFLG .EQ. 1)GO TO 80
       SALAT=DECON(ALAT,NALAT,MALAT,1)
       SSLAT=DECON(SLAT,NSLAT,MSLAT,1)
       SVLAT=DECON(VLAT,NVLAT,MVLAT,1)
       SALONG=DECON(ALONG,NALONG,MALONG,2)
       SSLONG=DECON(SLONG,NSLONG,MSLONG,2)
       SVLONG=DECON(VLONG,NVLONG,MVLONG,2)
       GO TO 66
 65    SALONG=DECON(ALONG,NALONG,MALONG,2)
       WRITE (MO,280) NALONG,MALONG,SALONG,THETA
       GO TO 80
 66    WRITE(MO,230) NHOUR,NMIN,NSEC,NALAT,MALAT,SALAT,TH,PITCH,NSLAT,
      1MSLAT,SSLAT,H,SLANT,THETA,THETAC,ALPHA,RBEAR,NVLAT,MVLAT,SVLAT,
      2RAD,DME,VARMAG,SENSE,WPAFB
       WRITE(MO.240) NALONG,MALONG,SALONG,NSLONG,MSLONG,SSLONG,NVLONG,
      1MVLONG,SVLONG,GCD
       IF (NFLAG.EQ.1) WRITE (MO,243)
       IF (NFLAG.EQ.2) WRITE (MO,244)
       WRITE (MO,245) REMARK
       GO TO 80
 70    WRITE(MO,250) NHOUR,NMIN,NSEC,ALAT,TH,PITCH,SLAT,H,SLANT,THETA,
      1THETAC,ALPHA,RBEAR,VLAT,RAD,DME,VARMAG,SENSE,WPAFB
       WRITE(MO.260) ALONG,SLONG,VLONG,GCD
       WRITE (MO,245) REMARK
 80    N=N+1
       IF (OUTOPT.LT.2) GO TO 90
       I=I+1
       IF (NFLAG.EQ.2)  ALONG=HOLD
       ALONG=ALONG+STEP
       IF(ALONG.GT. 180.0) ALONG=ALONG-360.0
       IF(ALONG.LT.-180.0) ALONG=360.0+ALONG
 90    IF(N.NE.18) GO TO 10
       WRITE(MO.270) ZETA
       IF (OUTOPT.GE.2) WRITE (MO,205)
       GO TO 5
1000   IF(OUTOPT .EQ. 3 .AND. PLTFLG .NE. 0) CALL PLOTND
       STOP
  110 FORMAT(I1,20A4)
  120 FORMAT(4I2,4F6.2,F5.0,3F3.0,2F4.1,I1,6A4,A1)
  200 FORMAT(1H1,25X,20A4)
 205   FORMAT (1H+,'ISO-ELEVATION POINTS')
 207   FORMAT (1H )
 210   FORMAT (1H ,'   TIME    ------AIRCRAFT-------   --------SATELLITE
```

```
     1-------  --ELEVATION--  ---AZIMUTH----  ---------VORTAC--------
     2AG  DIST.TO')
220  FORMAT(1H ,'  (ZULU)   LAT/LONG  T.H.  PITCH   LAT/LONG  HEIGHT
     1LANT RG  UNCORR   CORR  TRUE  RELATIVE  LAT/LONG  RADIAL  DIST
     2VAR   WPAFB')
221  FORMAT(1H0,'---FATAL ERROR---   VORTAC FIXES ARE INCOMPATIBLE WIT
     1 OUTPUT OPTIONS TWO OR THREE.')
222  FORMAT(1H0,'---FATAL ERROR---   OUTOPT GREATER THAN THREE')
223  FORMAT(1H0,'---FATAL ERROR---   NPDF GREATER THAN TWO')
224  FORMAT(1H0,'-FATAL ERROR---   INPUT LATITUDE GREATER THAN 90 DEGR
     1ES')
225  FORMAT(1H0,'-FATAL ERROR---ELEVATION ANGLE GREATER THAN 90 DEGREE
     1')
226  FORMAT(1H0,'-FATAL ERROR---   NEGATIVE SATELLITE ALTITUDE')
230  FORMAT(1H0, 2(I2,':'),I2,I3,'D',I2,'M',A2,F7.1,F6.1,I3,'D',I2,'M',
     1A2,F8.0,F9.0,2F8.1,F7.1,F8.1,I3,'D',I2,'M',A2,2F7.0,F6.1,A2,F8.0)
240  FORMAT (1H , 8X,I3,'D',I2,'M',A2,13X,I3,'D',I2,'M',A2,48X,I3,'D',
     1I2,'M',A2,'   GCD:',F7.0)
243  FORMAT (1H+,18X,'  1ST SOLUTION ')
244  FORMAT (1H+,18X,'  2ND SOLUTION ')
 245 FORMAT(1H+,54X,'REMARKS: ',6A4,A1)
250  FORMAT (1H0,2(I2,':'),I2,F9.2,F8.1,F6.1,2X,F8.2,2F9.0,2F8.1,F7.1,
     1F8.1,2X,F8.2,F8.0,F7.0,F6.1,A2,F8.0)
260  FORMAT (1H ,10X,F7.2,17X,F7.2,52X,F7.2,'   GCD:',F7.0)
 270 FORMAT(1H0,25X,20A4)
280  FORMAT(1H0,'0--- NO POINT EXISTS ALONG THE ',I4,' DEGREE',I3,' MIN
     1 ',A1,
     2 MERIDIAN AT WHICH THE ELEVATION ANGLE TO THE SATELLITE IS',F5.1,
     3' DEGREES',/)
     END
```

FILE PLMODS

```
C   CALLS PLT INITIALIZER
C   DRAWS AND LABELS EQUATOR
C   DRAWS LATITUDE OF SATELLITE & SAT SYMBOL
C       CALCULATIONS BASED ON X-AXIS OF 8.84375 INCHES=360 DEGREES LONG
C                 AND Y-AXIS OF 6.65625 = 180 DEGREES LATITUDE
C
C
        SUBROUTINE PLTSET(XPT,YPT)
        COMMON/PL/Y1DEG,Y0DEG,YAXIS,X1DEG,X180DG,X0DEG,XAXIS
        DIMENSION EQATRX(4),EQATRY(4),SATX(4),SATY(4)
        DATA EQATRX/0.0,8.84375,0.,1./
        DATA EQATRY/3.328,3.328,0.,1./
        DATA SATX(3),SATX(4)/0.0,1.0/
        DATA SATY/0.0,6.65625,0.0,1.0/
C
C
C   X AXIS RANGES FROM (-)60E TO (-)60E DEGREES LONGITUDE
        XAXIS=8.84375
        X1DEG=XAXIS/360.
        X180DG=X1DEG*120.0
        X0DEG=X1DEG*300.0
C
C    Y-AXIS RANGES FROM (-)90S TO 90N DEGREES LATITUDE
        YAXIS=6.65625
        Y1DEG=YAXIS/180.
        Y0DEG=Y1DEG*90.0
C   CONVERT ARGUMENTS TO INCHES FOR PLOTTING
C
        SX=XINCHS(XPT,YPT,YINCHS)
        SY=YINCHS
C
C
C    CALL REQUIRED INIT ROUTINE
        CALL PLOTST
C    DRAW EQUATOR
        CALL LINE(EQATRX,EQATRY,2,1,0,4)
C    LABEL EQUATOR
        CALL SYMBOL(0.5,Y0DEG+.05,.12,7HEQUATOR,0.0,7)
C
C    DRAW SAT LATITUDE
        SATX(1) = SX
        SATX(2) = SX
        CALL LINE(SATX,SATY,2,1,0,4)
C    DRAW SATELLITE
        CALL SYMBOL(SX,SY,.2,10,90.0,-1)
C    LABEL GRAPH
        CALL SYMBOL(.5,.2,.12,7HLES 8/9,0.0,7)
        CALL NUMBER(1.5,.2,.12,YPT,0.0,2)
        CALL SYMBOL(2.4,.2,.12,6HN OR S,0.0,6)
C***
        RETURN
        END
        FUNCTION XINCHS(X,Y,YINCHS)
        COMMON/PL/Y1DEG,Y0DEG,YAXIS,X1DEG,X180DG,X0DEG,XAXIS
```

```
C
C  COMPUTE INCH EQUIV OF X DEGREES
C      LEFT 2/3 OF X-AXIS
C
       IF(X .LT. 0.0 .AND. X .GT. -60.0) GO TO 10
       XTEMP=180. - ABS(X)
       IF(X .LT. 0.0)XINCHS=X180DG - (XTEMP*X1DEG)
       IF(X .GE. 0.0)XINCHS=X180DG + (XTEMP * X1DEG)
       GO TO 100
C
C  COMPUTE FAR RIGHT SIDE OF X-AXIS
C
10     XINCHS=X0DEG + (ABS(X) * X1DEG)
C
C  COMPUTE THE INCH EQUIV OF Y DEGREES
100    YINCHS=(Y + 90.) * Y1DEG
       RETURN
       END
       SUBROUTINE PLTISO(PLTX,PLTY,ITER)
       DIMENSION PLTX(362),PLTY(362)
       PLTX(ITER+1)=0.0
       PLTX(ITER+2)=1.0
       PLTY(ITER+1)=0.0
       PLTY(ITER+2)=1.0
       CALL LINE(PLTX,PLTY,ITER,1,-1,4)
       RETURN
       END
       INTEGER FUNCTION DECON(POSN,NDEG,MIN,NTYPE)
       INTEGER DECON
       IF(NTYPE.EQ.1) GO TO 4
       IF (POSN) 1,2,3
  1    DECON=2H E
       GO TO 8
  2    DECON=1H
       GO TO 8
  3    DECON=2H W
       GO TO 8
  4    IF(POSN) 5,6,7
  5    DECON=2H S
       GO TO 8
  6    DECON=1H
       GO TO 8
  7    DECON=2H N
     8 NDEG=  INT( ABS(POSN))
       MIN=  INT(( ABS(POSN)-NDEG)*60.0+.5)
       IF (MIN.NE.60) GO TO 9
       MIN=0
       NDEG=NDEG+1
  9    CONTINUE
       RETURN
       END
       FUNCTION CONDE(X)
       CONDE=1.6666667*X-.666667*IDINT(X)
       RETURN
```

```
      END
      FUNCTION SIND(X)
      SIND= SIN(X/57.2957795)
      RETURN
      END
      FUNCTION COSD(X)
      COSD= COS(X/57.2957795)
      RETURN
      END
      FUNCTION TAND(X)
      TAND= SIN(X/57.2957795)/ COS(X/57.2957795)
      RETURN
      END
      FUNCTION ASIND(X)
      IF(X.GE.1.0)GO TO 1
      IF(X.LE.-1.0)GO TO 2
      IF(X.EQ.0.0)GO TO 3
      ASIND=57.2957795*ATAN(X/SQRT(1.0-X**2))
      RETURN
    1 ASIND=90.0
      RETURN
    2 ASIND=-90.0
      RETURN
    3 ASIND=0.0
      RETURN
      END
      FUNCTION ACOSD(X)
      IF(X.EQ.0.0)GO TO 1
      IF(X.GE.1.0)GO TO 2
      IF(X.LE.-1.0)GO TO 3
      ACOSD=57.2957795*ATAN(SQRT(1.0-X**2)/X)
      IF(X.LT.0.0)ACOSD=ACOSD+180.0
      RETURN
    1 ACOSD=90.0
      RETURN
    2 ACOSD=0.0
      RETURN
    3 ACOSD=180.0
      RETURN
      END
      FUNCTION ATAND(X)
      ATAND=57.2957795* ATAN(X)
      RETURN
      END
```

SUBROUTINE AZI

```
      SUBROUTINE AZI(A,B,ALAT,SLAT,DELTA,ALPHA)
      R=57.2957795
C        CHECK DELTA TO PREVENT DIVISION BY ZERO
      IF(DELTA.EQ.-360.0)DELTA=-359.9999
      IF(DELTA.EQ.-180.0)DELTA=-179.9999
      IF(DELTA.EQ.180.0)DELTA=179.9999
      IF(DELTA.EQ.360.0)DELTA=359.9999
C        CALCULATE TANGENT OF DELTA
      D=SIN(DELTA/(2.0*R))/COS(DELTA/(2.0*R))
C        COMPUTE SLAT & DELTA IN RADIANS
      RSLAT=SLAT/R
      RDELTA=DELTA/R
C        COMPUTE HALF ANGLES OF SUM & DIFFERENCE
      DIF=(A-B)/(2.0*R)
      SUM=(A+B)/(2.0*R)
C        COMPUTE SUM IN DEGREES
      DSUM=SUM*R
C        CHECK HALF SUM TO PREVENT DIVISION BY ZERO
      IF(DSUM.EQ.180.0)SUM=179.9999/R
      IF(DSUM.EQ.0.0)SUM=.0001/R
      IF(DSUM.EQ.90.0)SUM=SIGN(89.9999,DELTA)/R
C        FIND LATITUDE WHICH CORRESPONDS TO AZIMUTH OF 90 DEGREES
      S=SIN(RDELTA)*COS(RSLAT)
      S=ASIND(S)
      IF(S.EQ.-90.0)S=-89.9999
      IF(S.EQ.90.0)S=89.9999
      S=S/R
      ANGLE=SIN(RSLAT)/COS(S)
      ANGLE=ABS(ASIND(ANGLE))
C        CHECK THAT THE 90 DEGREE LATITUDE IS CORRECT
      DI=(ANGLE-SLAT)/(2.0*R)
      SU=(180.0-ANGLE-SLAT)/(2.0*R)
      HS=R*ATAN(SIN(DI)/SIN(SU)/D)
      HD=R*ATAN(COS(DI)/COS(SU)/D)
      AA=HS+HD
      COSINE=COSD(AA)
      ANGLEN=-ANGLE
      DI=(ANGLEN-SLAT)/(2.0*R)
      SU=(180.0-ANGLEN-SLAT)/(2.0*R)
      HS=R*ATAN(SIN(DI)/SIN(SU)/D)
      HD=R*ATAN(COS(DI)/COS(SU)/D)
      AA=HS+HD
      C=COS(AA)
      IF(ABS(COSINE).GT.ABS(COSD(AA)))ANGLE=-ANGLE
C        CHECK TANGENT OF DELTA TO PREVENT DIVISION BY ZERO
      IF(D.EQ.0.0)GO TO 1
      GO TO 2
    1 ALPHA=180.0
      IF(ALAT.LE.ANGLE)ALPHA=0.0
      GO TO 5
    2 HSUM=R*ATAN(SIN(DIF)/SIN(SUM)/D)
      HDIF=R*ATAN(COS(DIF)/COS(SUM)/D)
      ALPHA=ABS(HSUM+HDIF)
      IF(ALPHA.GT.90.0)ALPHA=180.0-ALPHA
```

```
      IF((ANGLE.GT.0.0).AND.(ALAT.LT.ANGLE))GO TO 4
      IF((ANGLE.LT.0.0).AND.(ALAT.GT.ANGLE))GO TO 4
      IF((DELTA.GT.0.0).OR.(DELTA.LT.-180.0))GO TO 3
      ALPHA=180.0+ALPHA
      GO TO 5
    3 ALPHA=180.0-ALPHA
      GO TO 5
    4 IF((DELTA.GT.0.0).OR.(DELTA.LT.-180.0))GO TO 5
      ALPHA=360.0-ALPHA
   5  RETURN
      END
```

SECTION IV

4.1 Introduction. This section consists of the user instructions on operating RAPP and the associated support programs. A brief description follows of the tasks and programs that can be run by the operator. Those tasks which are run by other tasks (and are therefore transparent to the user) are not included.

RAPP - RSX task which reads the data cards, looks up the ephemeris, and points the antenna at the satellite.

SCAN - RSX task which performs a raster scan of the sky about the nominal pointing.

HAWAII - RSX task which predicts hourly look angles for flight planning purposes.

ALASKA - RSX task which predicts periodic look angles for flight planning purposes.

WRITER - RSX task which prints out the actual antenna look angle information on the line printer.

DATA - RSX task which prints all the ephemeris data currently stored on the disk.

XLINK - RSX task which records satellite ranging and look angle information to assist Lincoln Laboratory in orbit fitting.

SATSPT - DOS program which computes the look angle to any satellite from any point on the earth's surface. This program is especially useful for satellites for which ephemeris data is not available. The satellite latitude, longitude, and altitude must be known.

RCA - DOS program which formats the ephemeris data into Baudot and punches it onto paper tape. The reformatted ephemeris is then used on the IBM 4 PI computer, located on the C135 aircraft used in flight testing.

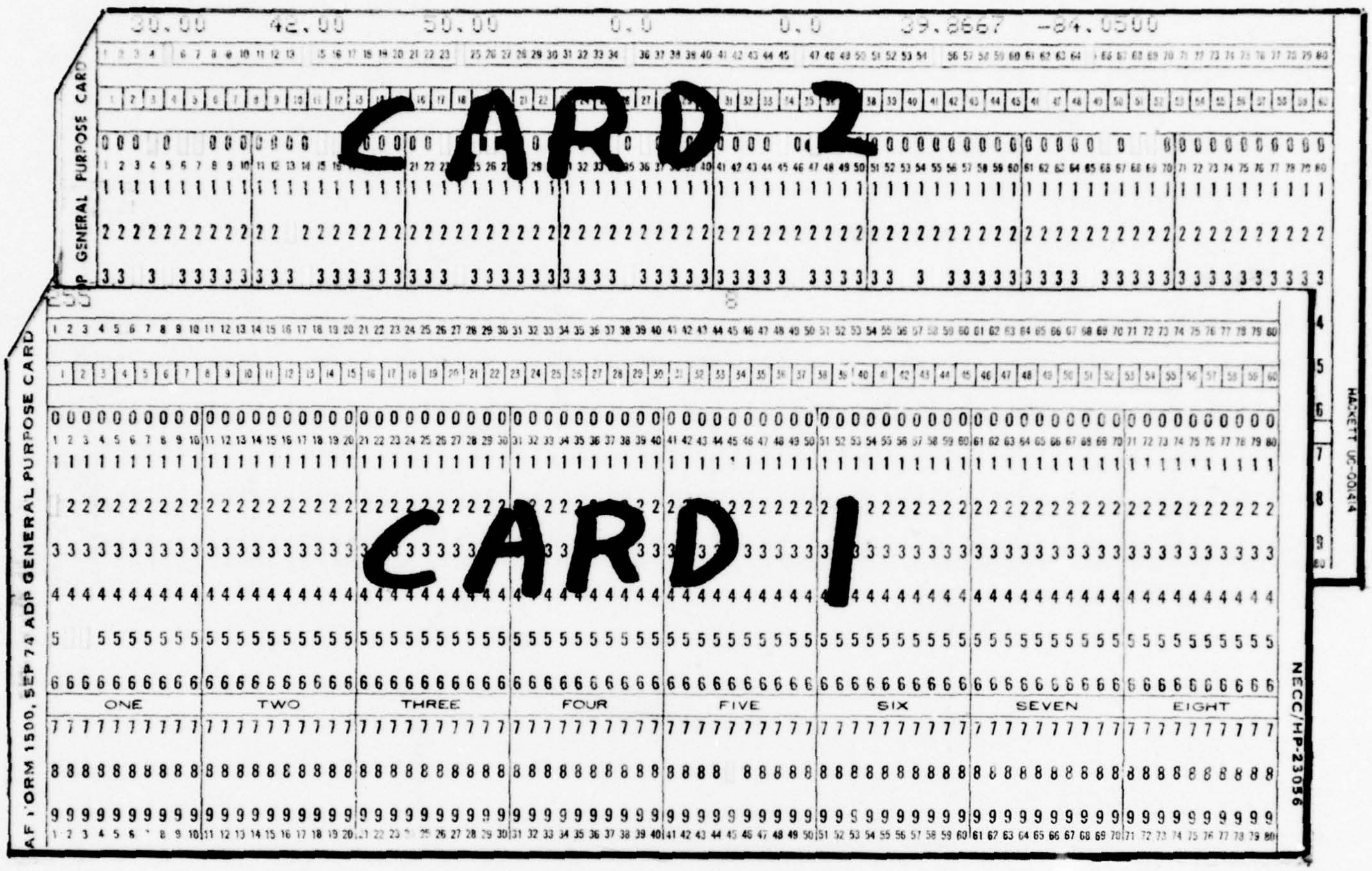
TRW - DOS program which punches the ephemeris data directly onto paper tape for use by the TRW Spread Spectrum Modem Processor, located in the Rooftop.

4.2 Running RSX Tasks. In the discussion which follows, it is assumed that: RSX has been booted, the operator has entered a User Identification Code (UIC) of 142, 4, and all tasks have been installed either manually or via the indirect install command "INS @ INSTL". It is further assumed that the system disk (DKØ) contains RAPP and all associated support programs, in addition to the ephemeris data.

4.2.1 Running RAPP. To run RAPP, the operator must prepare the data cards which contain the satellite and weather data. Figure 12 illustrates the data cards. When entering the Wright-Patterson location, note that northern latitudes and eastern longitudes are positive. In each case on data card 2, the numbers can contain up to four digits to the right of the decimal place.

Once the data cards have been prepared, the operator must enter the time-of-day into the computer, as it appears on the rubidium standard. It is important to enter the correct time of day, as RAPP uses computer time as a back-up whenever the WWVB receiver goes out of synch. After the card reader and line printer are brought on line, the operator may RUN RAPP.

Once RAPP has been run, and the ephemeris data printed on the line printer, the antenna is pointed at the desired satellite. Due to slight misalignments on the 10-foot dish, small biases must be added to (or subtracted from) the look angle. This is done from the VTØ5 terminal located in the Rooftop. The operator punches A±< bias term > or E±< bias term >, depending on whether azimuth or elevation biases are desired. A carriage return then enters the bias term into the computer. By observing



COLUMN NO.	CARD #	DATA
1-3	1	3-digit day of year
45	1	Satellite number
1-10	2	Barometric pressure in inches of mercury
11-20	2	Temperature in Fahrenheit
21-30	2	Relative humidity in percent
51-60	2	Wright-Patterson latitude
61-70	2	Wright-Patterson longitude

FIGURE 12. RAPP DATA CARDS

the spectrum analyzer located nearby, the operator can quickly peak up the K-band signal from the satellite.

4.2.2 Running SCAN. To run SCAN, the operator need only enter RUN SCAN. Task SCAN automatically suspends antenna pointing and scans the sky for the satellite. Upon completion, two minutes later, task SCAN resumes all antenna pointing tasks.

4.2.3 Running HAWAII. To run HAWAII, the operator simply prepares the data cards and types RUN HAWAII on the DECWRITER. Data card preparation consists of punching the day-of-year in columns 1-3, the satellite number (8 or 9) in column 6, and the desired longitude and latitude in columns 7-16 and 17-26 respectively. Note that northern latitude and eastern longitude are positive. When entering longitude, insure that the magnitude of the longitude is less than or equal to 180 degrees. Once a data card has been punched for each way-point on the proposed flight path, HAWAII can be run. A prediction of hourly look angles to the appropriate satellite, from 00:00 to 24:00Z, will be printed for each way-point. If a plot is desired, the output data should be spooled onto the disk, then transferred onto a DOS disk via FLX. Using the CALCOMP plotter on the 11/20, look angles can be plotted either as a function of time or as a function of position.

4.2.4 Running ALASKA. Data card preparation for task ALASKA is identical to that of task HAWAII. When ALASKA is run, the DECWRITER will query the operator for the specific time it is anticipated that the aircraft will arrive at a specific way-point. ALASKA will then print out the look-angle at that time, plus one, two, and three hours later. Like HAWAII, the output data can be spooled onto the disk, for future use.

4.2.5 Running WRITER. Like task SCAN, task WRITER can be run simply by typing RUN WRITER on the DECWRITER. Only run while the terminal is in active track and azimuth and elevation biases are zero, WRITER can be used to ascertain misalignments in the ten-foot dish. The pointing errors vary according to both the time of day and the accuracy of the ephemeris data. Therefore, WRITER needs to be run over a continuous 24-hour period, using several different orbit fits before permanent azimuth and elevation bias could be incorporated into RAPP.

4.2.6 Running DATA. Task DATA is run after the line printer has been placed on-line. DATA then prints all the ephemeris data presently stored on the disk.

4.2.7 Running XLINK. Task XLINK is run whenever ranging to the satellite is to be accomplished, in order to assist Lincoln Laboratory in orbit-fitting. Before running XLINK, a reel of magnetic tape must be mounted, placed on line, and initialized. The satellite must be placed in the correct mode, the Crosslink Ranging Receiver must be locked to the satellite, and the Rooftop terminal must be actively tracking the satellite.

Task XLINK is run by typing "RUN XLINK IS/RSI=4S" on the DECWRITER. This action runs XLINK, in one second, and reschedules it to run every four seconds. Upon running XLINK, the operator is queried as to which satellite is being used, current weather information (to allow for refraction correction), and other data required for orbit fitting. The required ranging data and pointing information will then be automatically sampled and recorded on mag tape every four seconds, during the first ten minutes of every hour. The mag tape file can be closed out when it is desired to complete data recording by setting Bit Zero of the 11/45 switch register to a "ONE". Task XLINK should then be CANcelled and ABOrted via the DECWRITER, and the mag tape removed for safekeeping.

4.3 Running DOS Programs. In the following discussion, it is assumed that DOS has been booted, and the operator has logged in under user number 142,4. DOS Version 9 is used for all DOS programs except SATSPT, which requires Version 8. Until a plotter driver is available under Version 9, Version 8 will have to be used if a plot is desired. Regardless of the version used, the system disk (DKØ) should contain all the RAPP support programs in addition to the ephemeris data.

4.3.1 Running PAUL. Program PAUL runs under DOS on the 11/45. First, the operator mounts the mag tape containing the ephemeris data on the mag tape unit. The mag tape unit is then brought on-line. The previous ephemeris data file on the disk, DAVEM.DAT, must be deleted prior to running PAUL. When PAUL is run, the ephemeris data is transferred from the mag tape onto the disk. When the mag tape stops moving, the transfer is complete. The operator types "control C" and "KILL" on the DECWRITER, and unloads the DOS disk.

The first step in getting the ephemeris data onto the RSX disk is to boot RSX on the 11/45 and mount the DOS disk which contains the ephemeris data as a foreign disk. The DOS disk is mounted using the following command string: "MOU DK1:/CHA=[FOR,DFA,ATCH,DCF]" followed by a carriage return. The old ephemeris data file DAVEM.DAT;1 on the RSX disk is deleted, and FLX invoked by the operator. The new RSX ephemeris file is copied onto the RSX disk via FLX by typing "DKØ:/IM=DK1:DAVEM.DAT" followed by a carriage return. The new ephemeris data may now be PIPed onto the other two RSX RAPP disks, following deletion of the respective old data files. Failure to delete the old file first could cause a new file to be formed (depending on the particular command string used) under DAVEM.DAT;2. Since RAPP uses the ephemeris located under DAVEM.DAT;1 it would use the old ephemeris.

4.3.2 Running SATSPT. SATSPT runs on the 11/20 under DOS Version 8, because the plotter is a peripheral on the 11/20 and requires a driver found only under Version 8. Data card preparation is more involved than with any other program or task, due to the many options available. In preparing data cards, the following conventions and limitations must be adhered to:

(a) All decimal places are implied. Do not punch decimal places on the data cards.

(b) All numerical data must be right-justified (must end in the right-most column of its field). The use of leading zeros is recommended.

(c) Default option (values assigned when the field is empty) is zero for numerics and blank for alpha-numerics.

(d) North latitude, west longitude, and west magnetic variation are positive by convention.

(e) Each data set must contain at least three cards:

(1) Output Option and Title Card

(2) Aircraft/Satellite Data Card(s)

(3) Sentinel Card

(f) Nose-up aircraft pitch is positive.

To prepare the data cards, use Figures 13-16 as a guide. Note that Figure 14 is used for preparing the Aircraft/Satellite data card for Output Options 0 or 1, while Figure 15 is used for Options 2 or 3.

To run SATSPT, insure that the card reader and line printer are on-line. In addition, as Output Option 3 is being used and a plot is being made, insure that the CALCOMP plotter is ready. Assign the card reader to device number 1, and type "RUN SATSPT" on the terminal.

4.3.3 Running RCA. RCA is run only on the 11/20, due to the availability of a high speed paper tape punch. Prior to running RCA, it must be verified that current ephemeris data is on the disk. If not, the data must be PIPed onto the disk from the 11/45 DOS disk which contains the latest ephemeris. To PIP the ephemeris from the 11/45 disk, type DAVEM.DAT/CO<DK1:DAVEM.DAT on the 11/20 terminal. It is imperative that the contiguous switch be used. Otherwise, errors will result when running RCA. Once this has been done, insure that RCA.LST (which may be left from the last time RCA was run) is deleted from the disk. To run RCA, simply

COL. NOS.	VARIABLE NAME	INPUT FORMAT	DESCRIPTION/REMARKS
1	OUTOPT	X	OUTOPT = 0: Non-iterative elevation and azimuth calculations with output in decimal format. OUTOPT = 1: Non-iterative elevation and azimuth calculations with output in degrees and minutes format. OUTOPT = 2: Program calculates aircraft latitudes which yield a specified elevation angle (iso-elevation line-of-position) over a longitude range specified by input data. Output format is degrees and minutes. OUTOPT = 3: Performs function described under OUTOPT = 2 for a range of elevation angles specified by the input data. An iso-elevation angle map may be drawn from the computed positions.
2-79	ZETA	Alpha-numeric	78 alpha-numeric characters printed as a header and trailer on each page. Whenever OUTOPT = 2 or 3, the word "ISO-ELEVATION" immediately precedes the contents of ZETA on the header and trailer lines.

Figure 13

INPUT DATA CARD NO. 1

COL. NOS.	VARIABLE NAME	INPUT FORMAT	DESCRIPTION/REMARKS
1-2	NHOUR	XX	Observation Hour.
3-4	NMIN	XX	Observation minute.
5-6	NSEC	XX	Observation second.
8	NPDF	X	NPDF = 0: All latitudes and longitudes in decimal format. NPDF = 1: All latitudes and longitudes in degrees and minutes format. NPDF = 2: Aircraft position to be determined by a VORTAC fix. VORTAC location and satellite subpoint in degrees and minutes format.
10-14	ALAT	±XX.XX -or- ±DDMM	If NPDF = 0: Aircraft latitude in decimal format. -or- If NPDF = 1: Aircraft latitude in degrees and minutes. If NPDF = 2: VORTAC station latitude in degrees and minutes. Converted to VLAT in program.
15-20	ALONG	±XXX.XX -or- ±DDDMM	If NPDF = 0: Aircraft longitude in decimal format. -or- If NPDF = 1: Aircraft longitude in degrees and minutes. If NPDF = 2: VORTAC station longitude in degrees and minutes. Converted to VLONG in program.
22-26	SLAT	±XX.XX -or- ±DDMM	If NPDF = 0: Satellite subpoint latitude in decimal format. -or- If NPDF = 1 or 2: Satellite subpoint latitude in degrees and minutes.
27-32	SLONG	±XXX.XX -or- ±DDDMM	If NPDF = 0: Satellite subpoint longitude in decimal format. -or- If NPDF = 1 or 2: Satellite subpoint longitude in degrees and minutes.
33-37	H	XXXXX.	Satellite height above subpoint (nautical miles)

Figure 14a

Input Data Cards 2 through K+1 (for K observations)
Output Option 0 or 1 Only

COL. NOS.	VARIABLE NAME	INPUT FORMAT	DESCRIPTION/REMARKS
38-40	HEAD	XXX.	Aircraft magnetic heading.
41-43	RAD	Blank -or- XXX.	If NPDC = 0 or 1. -or- If NPDF - 2: Magnetic bearing from VORTAC station to aircraft (radial).
44-46	DME	Blank -or- XXX.	If NPDF = 0 or 1. -or- If NPDF = 2: Distance from VORTAC station to aircraft (nautical miles)
47-50	VARMAG	±XX.X	If NPDF = 0 or 1: Magnetic variation at the aircraft location. If NPDF = 2: Magnetic variation at the VORTAC station. The mag. var. at the aircraft is assumed to be equal to the station's value.
51-54	PITCH	±XX.X	Aircraft pitch angle, usually displayed on a clinometer.
56-80	REMARK	Alpha-numeric	25 Alpha-numeric characters printed with each output record following the "REMARKS:" label.

Figure 14b

Input Data Cards 2 through K+1 (for K observations)
Output Option 0 or 1 Only

COL. NOS.	VARIABLE NAME	INPUT FORMAT	DESCRIPTION/REMARKS
1-6	NHOUR, NMIN, NSEC	XXXXXX	Any 6-digit identification number not beginning with 99.
8	NPDF	X	If NPDF = 0: Latitude, longitudes, and longitude increment are in decimal format. If NPDF = 1: Latitude, longitudes and longitude increment are in degrees and minutes.
9-14	ALAT	±XX.XX -or- ±DDMM	If NPDF = 0: Aircraft longitude increment in decimal format. -or- If NPDF = 1: Aircraft longitude increment in degrees and minutes. Note: ALAT is converted to STEP within the program.
15-20	ALONG	±XXX.XX -or- ±DDDMM	If NPDF = 0: Initial aircraft longitude in decimal format. -or- If NPDF = 1: Initial aircraft longitude in degrees and minutes.
22-26	SLAT	±XX.XX -or- ±DDMM	If NPDF = 0: Satellite latitude in decimal format. -or- If NPDF = 1: Satellite latitude in degrees and minutes.
27-32	SLONG	±XXX.XX -or- ±DDDMM	If NPDF = 0: Satellite longitude in decimal format. -or- If NPDF = 1: Satellite longitude in degrees and minutes.
33-37	H	XXXXX.	Satellite height above subpoint (nautical miles).
38-40	HEAD	XXX.	Number of aircraft longitude iterations for a given elevation angle. Converted to ITER in program.
44-46	DME	Blank -or- XXX.	If OUTOPT = 2. -or- If OUTOPT = 3: Number of elevation angle iterations. Converted to NELEV in program.

Figure 15a

Input Data Card No. 2, Output Option 2 or 3 Only

COL. NOS.	VARIABLE NAME	INPUT FORMAT	DESCRIPTION/REMARKS
47-50	VARMAG	Blank -or- ±XX.X	If OUTOPT = 2 -or- If OUTOPT = 3: Elevation angle increment. Converted to ELSTEP in program.
51-54	PITCH	±XX.X	If OUTOPT = 2: Elevation angle. If OUTOPT = 3: Initial elevation angle. Converted to THETAT in program. Negative elevation angles are acceptable. Must always be $<90^{o}$.
55	PLTFLG	X	If PLTFLG = 0: No plot. If PLTFLG = 1: Plot only. If PLTFLG = 2: Plot plus listing of values.
56-80	REMARK	Alpha-numeric	25 Alpha-numeric characters printed with each output record following the "REMARKS:" label.

Figure 15b

Input Data Card No. 2
Output Option 2 or 3 Only

COL. NOS.	VARIABLE NAME	INPUT FORMAT	DESCRIPTION/REMARKS
1-2	NHOUR	99	Indicates end of data applicable to the most recent OUTOPT and Title Card. The next card should be either an end-of-file (7-8-9) card or another OUTOPT and Title Card (input data card no. 1) and associated data card(s).
56-80	REMARK	Alpha-numeric	25 alpha-numeric characters. Not printed.

Figure 16

End-of-Data Card

type RUN RCA, and respond with the desired day-of-year when queried by the terminal. RCA will spool the desired ephemeris onto the disk as RCA.LST, reformat the data in the BAUDOT format, and punch the reformatted data on the high-speed paper tape punch. RCA.LST can then be PIPed onto the line printer prior to deletion.

4.3.4 Running TRW. Program TRW is run much the same as program RCA; after insuring that current ephemeris data is on the disk, simply type RUN TRW on the terminal. Respond with the proper day-of-year, and TRW will punch a paper tape directly from the disk. Since no re-formatting is involved, no files are spooled onto the disk. The paper tape may be verified simply by PIPing it from the paper tape reader onto the line printer.

4.4 Fault-Finding Chart. A brief discussion follows of the various faults which could prevent a task or program from executing correctly.

Program/Task	Problem	Solution
ALASKA	DAVEM.DAT file locked.	Run PIP and unlock file.
	Error in entering time.	Re-enter correct time.
ASMINT	Error in sampling WWVB time-of-day.	Turn on WWVB receiver.
DATA	DAVEM.DAT file locked.	Run PIP and unlock file.
	No output on line printer.	Reset line printer and place on-line.
DOIT	Antenna does not point.	Turn on Antenna Control Interface Unit. Enable antenna pedestal in Rooftop.
FLX	Insufficient contiguous space on disk for DAVEM.DAT.	Run Disk Compression Utility (DCU).
HAWAII	DAVEM.DAT file locked.	Run PIP and unlock file.
PAUL	File transfer error.	Delete old DAVEM.DAT prior to running PAUL.

RAPP	DAVEM.DAT file locked.	Run PIP and unlock file.
	Incorrect pointing of antenna.	Enter correct time-of-day as it appears on the rubidium standard.
RTLOOP	Incorrect pointing of antenna.	Antenna bias term as input from VT05 is out of range. Re-enter correct bias term.
RCA	File transfer error.	Delete old RCA.LST file prior to running PAUL.
SCAN	Antenna does not point.	Turn on Antenna Control Interface Unit.
WRITER	No output on line printer.	Reset line printer and place on-line.
	Incorrect data printed.	Turn on Antenna Control Interface Unit.
XLINK	No output to mag tape.	Initialize mag tape. Insert write enable ring on tape reel.
	Incorrect data recorded on tape.	Insure that Antenna Control Interface Unit, WWVB receiver, and Crosslink Ranging Receiver and Crosslink Ranging Interface are powered up.

4.5 Dictionary of Computer Symbols

A — Intermediate term used in several tasks and subroutines. In SCAN, A is used in generation of the azimuth bias offset. In subroutine HINIT, A is used in a latitude conversion from degrees to radians. In subroutines CONVER and AEOUT, A is used as the antenna bias correction term. In task INPUT, A is used as the upper (floating) word of the input from the Crosslink Ranging Receiver Interface.

ABIAS — Azimuth bias term used in task SCAN. Its value is passed to task SCANR to compute the new desired azimuth pointing.

ADDØ — Used in subroutine TOD, as the Control and Status Register address of the WWVB receiver interface with the 11/45 computer.

ADD4 — Used in subroutine TOD, as the input buffer address of the WWVB receiver interface with the 11/45 computer.

AK1-AK3 — Provided by subroutines DATAID and HDATA, used in calculating the refractivity constant.

AL — Intermediate azimuth value used in task SCAN.

ALAT — Aircraft or Avionics Lab latitude, corrected for the earth's oblateness. Calculated in subroutines DATAID and HDATA; also used in subroutines INITAL and HINIT as an intermediate calculation in determining the sine and cosine of the latitude.

ALLAT — Aircraft or Avionics Lab latitude, as input from the input data cards in subroutines DATAID and HDATA.

ALLNG — Aircraft or Avionics Lab longitude, as input from the input data cards in subroutines DATAID and HDATA.

ALRAD — Approximate radius (in meters) from the center of the earth to aircraft or Avionics Lab position. Computed in subroutines DATAID and HDATA.

AMIN — Floating point "minutes" term of time-of-day. Used in task WRITER.

AN — Surface refractivity of the atmosphere. Computed in subroutines DATAID and HDATA.

ANTFLG — Flag set by task DOIT once every five seconds. When the set condition is sensed by task RTLOOP, the VT-Ø5 display is sampled for any input changes desired. ANTFLG is then reset by RTLOOP.

AXXA — Absolute value of variable XXA.

AXYA — Projection of variable XXA and YYA onto the X-Y plane. Used in subroutines ANTENA, HANTEN, and PANTEN.

AYYA — Absolute value of variable YYA.

AZ — Intermediate value of azimuth used in task SCAN.

AZBIAS — Antenna azimuth bias term. Initialized in subroutines DATAID and HDATA. Computed in subroutine AEOUT, and used in subroutines ANTENA, HANTEN, and PANTEN.

AZI — Computed antenna azimuth. Calculated in subroutines ANTENA, HANTEN, and PANTEN. Used in computing the azimuth error in subroutine PNT.

AZIIN — Actual antenna azimuth, as input from the antenna servos by subroutine STATE. Used in subroutine PNT to calculate the azimuth command.

AZZA — Absolute value of variable ZZA.

B — Used in task INPUT as a temporary calculation in computing the path delay to the satellite.

CHKT — Difference (in seconds) between WWVB and 11/45 computer time. Computed in subroutine CTIME.

CLAT Cosine of aircraft or Avionics Lab latitude. Computed in subroutines INITAL and HINIT. Used in subroutines ANTENA, HANTEN and PANTEN.

CLLAT Aircraft or Avionics Lab latitude in radians. Computed in subroutines INITAL and HINIT.

CLNG Cosine of aircraft or Avionics Lab longitude. Used in subroutines HINIT, INITAL, ANTENA, HANTEN, and PANTEN.

CON1 In task INPUT, used as a constant in scaling the range to the satellite. In subroutine TOD, used as a constant to convert milliseconds to seconds. In subroutine PNT, used to add 64 to the pointing command. Paragraph 2.4.4.3.2 explains why this is done.

CON2 In task INPUT, used as a constant in scaling the range to the satellite. In subroutine TOD, used as a constant to convert seconds to tens of seconds. In subroutine PNT, used as a constant to check if the difference between the measured and calculated range is less than -10 degrees. If so, a limit of -10 degrees is placed on the pointing command.

CON3 In subroutine PNT, used to subtract 360 degrees from the azimuth if the azimuth is larger than 360 degrees.

CON4 In subroutine PNT, used to add 59 degrees to both input and output azimuths. This rotates the azimuth reference as mentioned in paragraph 2.4.4.3.2.

CON5 In subroutine PNT, used as a constant to check if the difference between the measured and calculated azimuth is greater than ten degrees. If so, a limit of ten degrees is placed on the pointing command.

COUNT	Used as a counter in task DOIT to allow updating of antenna look-angle every five seconds.
DABIT	D/A converter scale factor used in subroutine PNT. Equal to .03125 degrees per bit.
DT	One-half satellite position update period (2.5 seconds) used in subroutines CTIME and HTIME.
DTC	No longer used, but still appears in subroutines DATAID and HDATA.
EBIAS	Elevation bias term used in task SCAN. Used to effect the elevation stepping of the raster scan.
EL	Intermediate value of elevation used in task SCAN.
ELBIAS	Antenna elevation bias term. Initialized in subroutines DATAID and HDATA. Computed in subroutine AEOUT and used by subroutines ANTENA, HANTEN, and PANTEN.
ELEIN	Actual antenna elevation. Input by subroutine STATE and used in subroutine PNT to generate elevation error command.
ELEV	Computed antenna elevation. Calculated in subroutines ANTENA, HANTEN, and PANTEN. Used in subroutine PNT to generate elevation error command.
ERA	Antenna azimuth error. Difference between actual and computed antenna azimuth. Calculated in subroutine PNT.
ERE	Antenna elevation error. Difference between actual and computed antenna elevation. Calculated in subroutine PNT.
F1-F20	Dummy floating point variable common block. Used for present and future expansion.
HOLD	Intermediate variable used in subroutine PNT. Used to hold the antenna servo commands to the azimuth and elevation, in turn.

HOUR	Hours term from the time-of-day, used in task WRITER.
I	Used in subroutine CONVER as an intermediate input term to compute the azimuth or elevation input bias. Also used in subroutine AEOUT as an intermediate calculation of the range to the satellite.
IARAY(N)	Nth element of the buffer used in reading the ephemeris data off the disk. Used in subroutines DATAID and HDATA.
IDAY	Desired day-of-the-year. For instance, 14 February = Day 45. Input on data cards by subroutines DATAID and HDATA, and by tasks DATA and DATAIL.
IDOT	Used by subroutine CONVER to flag a decimal point occurring in the antenna pointing bias term.
IEOF	A flag set in subroutines DATAID and HDATA and tasks DATA and DATAIL if the desired ephemeris data cannot be located on the disk. Passed to tasks RAPP, ALASKA, or HAWAII to allow printing of an error message when this occurs.
IFLG	Flag used in inputting correct ephemeris data block. Set when LES 8 ephemeris has been located. Used in subroutines DATAID and HDATA, and tasks DATA and DATAIL.
IHOUR	Integer representation of variable HOUR. Used in task WRITER.
INTFLG	Flag used to signal task SCAN that task SCANR has finished running and exited.
IODAY	Previous (old) value of day-of-the-year (IDAY-1).Used in subroutine HDATA, where IDAY is being incremented automatically.
IOUNT	FORTRAN integer counterpart of COUNT. Initialized in tasks RAPP, ALASKA, and HAWAII; variable COUNT must have an integer name in FORTRAN.

ISEC	Integer representation of variable SEC. Used in task WRITER.
ISIGN	Used in subroutine CONVER to flag a negative antenna bias term.
ITIME	Hours term of time-of-day, as entered from the DECWRITER in task ALASKA.
IVAR	Dummy variable used in the DEFINE FILE statement of subroutine DATAID and tasks DATA, DATAIL, ALASKA, and HAWAII.
I1-I20	Dummy integer variable common block. Used for present and future expansion.
J	Intermediate bias term used in subroutine AEOUT to calculate the slant range to the satellite.
JDAY	Actual value of the day-of-the-year as contained in the ephemeris data on the disk. Input in subroutines DATAID and HDATA and tasks DATA and DATAIL.
JFLG	Flag used in inputting correct ephemeris data block. Set when the A.M. buffer of the appropriate satellite has been located. Used in subroutines DATAID and HDATA, and tasks DATA and DATAIL.
JJ	Error return flag used in all CALL RUN statements. Also used in subroutine CONVER to compute the antenna bias term.
JT	Time-of-day in hours, used in task HAWAII.
JTIME	Minutes term of time-of-day, as entered from the DECWRITER in task ALASKA.
J1-J6	Dummy input variables buffer used in subroutine AEOUT. Utilizing a 6A1 format, requests can be made via this block for: pointing information, adding or subtracting of antenna pointing biases, or requests can be made for changing satellites.
K	No longer used.

KK	Counter used to mark data location during direct disk access of the ephemeris data. Used in subroutines DATAID and HDATA, and tasks DATA and DATAIL.
KOUNT	Used in subroutine CONVER as a counter for the five-digit azimuth or elevation bias term.
LES	Desired satellite number. Equals 8 for LES 8; 9 for LES 9. Input and used in subroutines DATAID, HDATA, and AEOUT, and tasks DATA and DATAIL.
LLES	Actual satellite number, as contained in the ephemeris data on the disk. Used in subroutines DATAID and HDATA.
MIN	Integer representation of variable AMIN. Used in task WRITER.
N	Used in inputting azimuth or elevation biases in subroutine CONVER. N=A implies an azimuth bias follows. N=E implies an elevation bias follows.
NUM	Integer representation of the floating point antenna bias term used in subroutine CONVER.
PP	Atmospheric pressure in inches of mercury. Input on data card in subroutines DATAID and HDATA.
PW	Partial water vapor pressure calculated from the weather data. Computed in subroutines DATAIL and HDATA.
R	Number of degrees in a radian (=57.2457795). Used throughout the Rooftop Antenna Pointing Program for conversions.
RAD	Same as variable R.
RH	Relative humidity in per cent. Input on data card in subroutines DATAID and HDATA.
RLAT	Avionics Lab or aircraft latitude in radians. Used in subroutines DATAID and HDATA.

SDBIT	Synchro to digital (S/D) converter scale factor. Equal to .00549325 degrees per bit. Used in subroutine STATE.
SEC	Secondsterm of time-of-day. Used in task WRITER.
SLAT	Sine of Avionics Lab or aircraft latitude. Computed in subroutines INITAL, HINIT, ANTENA, HANTEN, and PANTEN.
SLNG	Same as SLAT, except for longitude.
SW	"Switch" used to flag a return from subroutine PNT. When SW=2, both azimuth and elevation calculations have been completed, and PNT will execute a RETURN.
T	11/45 computer time, in seconds after midnight,GMT. Input in task RAPP to determine whether A.M. or P.M. ephemeris data buffer is to be loaded first.
TC	Air temperature in degrees Celcius. Calculated from Fahrenheit temperature in subroutines DATAID and HDATA.
TEST1	With a value of 270 degrees, TEST1 represents the lower limit of the forbidden antenna azimuth zone, when pointing is from the Avionics Lab. Used in subroutines ANTENA and PANTEN.
TEST2	At 302 degrees, TEST2 represents the upper limit of the forbidden antenna azimuth zone, when pointing from the Avionics Lab. Used in subroutines ANTENA and PANTEN.
TEST3	At 288 degrees, TEST3 represents the approximate center of the forbidden antenna azimuth zone, when pointing from the Avionics Lab. Used in subroutines ANTENNA and PANTEN.
TET	11/45 computer time, in seconds after midnight, GMT. Used in subroutine CTIME.
TETIM	Time-of-day in seconds after midnight. Input from WWVB in subroutine TOD for task RAPP, or else computed in tasks ALASKA and HAWAII.

TETI3	Scaled time-of-day, as used with the ninth order polynomials for look-angle calculation. Computed in subroutines CTIME and HTIME.
TF	Final time in seconds, of the A.M. or P.M. (GMT) time period. Equals 43200 in A.M. (corresponds to 1200Z). equals 86400 in P.M. (corresponds to 2400Z). Calculated in subroutines CTIME and HTIME, and used in subroutine DATAID.
TFAR	Air temperature in degrees Fahrenheit. Input on data cards by subroutines DATAID and HDATA.
TI	Initial time in seconds, of the A.M. or P.M. (GMT) time period. Equals 0 in A.M. (corresponds to 0000Z), equals 43200 in P.M. (corresponds to 1200Z). Calculated in subroutines CTIME and HTIME, and used in subroutine DATAID.
TK	Air temperature in degrees Kelvin. Calculated from Fahrenheit temperature in subroutines DATAID and HDATA.
TMID	Mid-time in seconds, of the A.M. or P.M. (GMT) time period. Equals 2160 in A.M. (corresponds to 0600Z), equals 64800 in P.M. (corresponds to 1800Z). Calculated in subroutines CTIME and HTIME, and used in subroutine DATAID.
TWV	Constant used in refractivity calculation. Used in subroutines DATAID and HDATA.

WORD	Two-word location used in assembler language subroutine TOD. Used as a buffer for the two-word time-of-day input from the WWVB interface.
X, Y, Z	Temporary value of XCA, YCA, or ZCA respectively. Used in subroutines DATAID and HDATA, and task DATA. Effects the placing of the A.M. and P.M. buffers in the correct memory locations. Z is also used in subroutine AEOUT as an intermediate calculation of the slant range to the satellite.
XAL, YAL, ZAL	X, Y, or Z component respectively of the Avionics Lab or aircraft position in geocentric coordinates. Computed in subroutines INITAL and HINIT.
XC(N), YC(N), ZC(N)	Nth element of the X, Y, or Z axis satellite position polynomial. Input from the appropriate A.M. or P.M. ephemeris buffer via subroutines CTIME and HTIME.
XCA(N), YCA(N), ZCA(N)	Nth element of the A.M. X, Y, or Z axis satellite position polynomial. Transferred from XC(N), YC(N) or ZC(N) respectively in subroutines DATAID and HDATA.
XCB(N), YCB(N), ZCB(N)	Nth element of the P.M. X, Y, or Z axis satellite position polynomial. Transferred from XC(N), YC(N), or ZC(N) respectively in subroutines DATAID and HDATA.
XSAT, YSAT, ZSAT	X, Y, or Z component respectively of the satellite position in geocentric coordinates. Used in subroutines ANTENA, HANTEN, and PANTEN.
XSATC, YSATC, ZSATC	X, Y, or Z term respectively of a correction applied to each component of the satellite position. Currently equals zero in subroutines where it is used - ANTENA, HANTEN, and PANTEN.

XX, YY, ZZ — X, Y, or Z component respectively of the vector from the Avionics Lab or aircraft to the satellite, in geocentric coordinates. Used in subroutines ANTENA, HANTEN, and PANTEN.

XXA, YYA, ZZA — XX, YY, or ZZ respectively in geodetic coordinates. Used in subroutines ANTENA, HANTEN, and PANTEN. ZZA is also used later in the same subroutines to determine the sign on the elevation reference calculation.

REFERENCES

1. "Operation, Installation, and Maintenance Instructions for the Digital Data Modem Group OM-53(XA-1)/ASC-22" for Contract F33615-73-C-4141, TRW Defense and Space Systems Group, Redondo Beach, California.

2. "Ka-Band Rooftop Antenna Subsystem Technical Manual" for Contract F33615-74-C-4012, Raytheon Company, Sudbury, Massachusetts.

3. "American Ephemeris and Nautical Almanac", U.S. Government Printing Office.

4. "Interface Specification for Spread Spectrum Modem/Processor Subsystem", Report No. 101-AFAL, TRW Systems Group, Redondo Beach, Calif.

5. "Time Standard System for Directing Satellite Tracking Antennas", Technical Report AFAL-TR-76-145, Engineering Experiment Station, Georgia Institute of Technology, Atlanta, Georgia.

6. "SATSPT: An Aircraft/Satellite Relationship Program with Iso-elevation Line Calculation Options", Lt Joseph F. Knapick, Jr., 4950th Test Wing, Wright-Patterson AFB, Ohio.

7. "Satellite Configuration for Crosslink Receiver Ranging", Technical Memorandum dated 26 August 1976, G.W. Sellers, Lincoln Laboratory.

8. "A General Discussion of the Hardware Aspects of the Crosslink Ranging Method in the Air Force and Navy Terminal", Technical Memorandum dated 12 February 1976, G.W. Sellers, Lincoln Laboratory.

9. "Software for Satellite Ranging up to PEP Input", Technical Memorandum dated 24 February 1976, G.W. Sellers, Lincoln Laboratory.